

(19) 日本国特許庁 (J P)

(12) 特 許 公 報 (B 2)

(11) 特許番号

第2593146号

(45) 発行日 平成 9 年 (1997) 3 月 26 日

(24) 登録日 平成 8 年 (1996) 12 月 19 日

(51) Int. Cl. <sup>8</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/163			G 0 6 F 15/16	3 2 0 A
13/38	3 3 0		13/38	3 3 0 Z
			15/16	3 2 0 P

発明の数 1 (全 32 頁)

(21) 出願番号	特願昭61-154955	(73) 特許権者	999999999
(22) 出願日	昭和61年(1986) 7 月 1 日		ユナイテッド テクノロジーズ コーポ レーション
(65) 公開番号	特開昭62-70964		アメリカ合衆国 コネティカット
(43) 公開日	昭和62年(1987) 4 月 1 日		06101 ハートフォード ファイナンシ ャル プラザ 1 番地
(31) 優先権主張番号	7 5 0 9 2 3	(72) 発明者	ピーター マーク アサナス
(32) 優先日	1985年 7 月 1 日		アメリカ合衆国 コネティカット
(33) 優先権主張国	米国 (U S)		06074 サウス ウィンザー アーサー ロード 8-3 番地
審判番号	平7-19103	(74) 代理人	弁理士 伊東 忠彦
		合議体	
		審判長	木屋野 忠
		審判官	菅野 嘉昭
		審判官	松野 高尚

最終頁に続く

(54) 【発明の名称】 データハブ

1

(57) 【特許請求の範囲】

【請求項 1】 各々が個々に制御信号線を介して及び並列アドレス/データ信号線を介して個々のポートでハブに接続された複数のデータソース/転送先ユニット間のデータ流れを制御するデータハブであって、該データハブはマスタソース/転送先ユニットから供給される少なくとも一対のアドレスポインタ入力制御信号ワード、ワード計数入力制御信号ワード、及びチャンネルイネーブル入力制御信号ワードにตอบสนองして、上記入力制御信号ワードは上記ハブによってなされるソースとしてのデータソース/転送先ユニット内の一又はそれ以上のメモリ位置からの一又はそれ以上の予定されたデータ信号ワード転送を示し、上記ソースとしてのデータソース/転送先ユニットはチャンネルイネーブル入力制御信号ワードによって上記データハブにおいてイネーブルとされるチャン

2

ネルにおけるチャンネルソースユニットとして使用され、上記一又はそれ以上の予定されたデータ転送は上記ソースとしてのソース/転送先ユニットから転送先ユニットとして使用される転送先としてのイネーブルとされたチャンネルデータソース/転送先ユニット内の一又はそれ以上のメモリ位置へ上記データハブによってなされ、上記一又はそれ以上のチャンネルソースユニットメモリ位置の第 1 のものはソースアドレスポインタ入力制御信号ワードによって示され残りのものは上記ワード計数に等しいいくつかの連続する上記ソースアドレスポインタ入力制御信号ワードの増加又は減少によって示され、上記一又はそれ以上の転送先メモリ位置の第 1 のものはデータ転送先アドレスポインタ入力制御信号ワードによって示され残りのものは上記ワード計数に等しいいくつかの連続する上記転送先アドレスポインタ入力制御信

10

号ワードの増加又は減少によって示され、上記ハブは更に、上記転送先としてのイネーブルとされたチャンネルデータソース／転送先への後の転送のために上記ソースとしてのイネーブルとされたチャンネルソース／転送先から転送されたデータを一時的に記憶する記憶手段よりなり、上記データハブは：

上記ソース及び転送先アドレスポインタ入力信号ワード、上記ワード計数及び上記チャンネルイネーブル入力制御信号ワードにตอบสนองして上記イネーブルにされたデータソース／転送先ユニットに対してソースデース転送要求出力信号を供給し次にその代りに上記ソースとしてのイネーブルにされたデータソース／転送先ユニットからソースバス許可信号を受け取る制御手段であって、該制御手段は上記ソースバス許可信号にตอบสนองして上記ソースアドレスポインタ入力制御信号ワードを出力信号として上記ソースとしてのイネーブルにされたデータソース／転送先ユニットに供給し、上記記憶手段は同時にソースデータ信号ワードを上記ソースアドレスポインタ出力制御信号ワードによって示されるような上記ソース／転送先ユニット内のメモリ位置からソースデータ信号ワードを受けとって一時的に記憶し、上記制御手段はそこで上記転送先としてのデータソース／転送先ユニットに転送先データ転送出力要求信号を供給し次にその代りに上記転送先としてのデータソース／転送先ユニットから転送先バス許可信号を受け取り、上記制御手段は上記転送先バス許可信号にตอบสนองして前記転送先アドレスポインタ入力制御信号ワードを転送先アドレスポインタ出力制御信号ワードとして上記転送先としてのデータソース／転送先ユニットに供給し同時に上記記録手段から上記転送先アドレスポインタ出力制御信号によって示されるような上記転送先としてのソース／転送先ユニット内のメモリ位置への上記データ信号ワードの転送を行い、もって上記ソースとしてのデータソース／転送先と上記転送先としてのデータソース／転送先とでメモリを共有することなくデータ転送がなされ、上記制御手段は上記ソース及び転送先アドレスポインタ出力信号を増加又は減少させ上記ワード計数信号を減少させ上記ワード計数信号がゼロに等しくなるまで上記ソースとしてのイネーブルにされたチャンネルのデータソース／転送先ユニットから上記転送先としてのデータソース／転送先ユニットへのデータの転送を周期的に繰り返す、制御手段を特徴とするデータハブ。

【請求項2】上記制御手段は更に上記記憶手段への非同期入来直接直列データ信号ワード転送に及び上記複数のデータソース／転送先ユニットの内の選択された一つからの非同期直列データ転送要求信号ワードにตอบสนองし、上記制御手段はそれにตอบสนองして予定されたデータ信号ワード転送を低位優先転送カテゴリに分類し非同期入来直接直列データ信号ワード転送を中位優先転送カテゴリに分類し非同期直列データ転送要求信号ワードを上位優先転

送カテゴリに分類し、上記制御手段は上記優先カテゴリを保ちつつ優先付けされたデータ転送要求出力信号ワードを供給し、上記制御手段は上位優先データソース／転送先からの上記上位優先非同期直列データ転送要求信号ワードにตอบสนองしてイネーブル信号を供給し、該イネーブル信号は上記上位優先データソース／転送先が直ちに他のデータソース／転送先から／ヘデータ信号を送信／受信することを許容し、上記制御手段は又上記記憶手段への上記中位優先非同期入来直接直列データ信号ワード転送にตอบสนองして上位優先非同期直列データ転送要求信号ワードによって表現される全ての転送が完了した後に上記中位優先非同期入来直線直列データ信号ワード転送を行い、それは次にその代りとして上記中位優先データソース／転送先ユニットから転送先バス許可信号を受けるために転送先バス要求信号を転送先ユニットとして使用される転送先としての中位優先データソース／転送先ユニットへ供給することによってなされ、上記制御手段は上記転送先バス許可信号が供給された後に上記記憶手段から上記転送先としての中位優先ソース／転送先ユニット内のメモリアドレスへのデータ信号転送を行い、上記制御手段は上記低位優先の予定されたデータ信号ワード転送にตอบสนองし、上位優先非同期直列データ転送要求信号ワードに関する全ての転送及び全ての中位優先データ信号ワード転送が完了した後に一又はそれ以上の低位優先データ信号ワード転送を行うこと

を更に特徴とする請求項1に記載のデータハブ。

【発明の詳細な説明】

産業上の利用分野

本発明はマルチプロセッサシステム及び分散プロセッサシステムに係り、特に分散プロセッサ間及びマルチプロセッサ間のデータ転送手段に関する。

従来の技術及びその問題点

分散処理のためのマルチプロセッサシステムアーキテクチャはプロセッサ間で共用される共通メモリを度々使用する。これはマルチプロセッサシステムでは複数ポートメモリ、あるいはクロスバススイッチ、タイムシェア共通バス、あるいは二重バス構造を含むアーキテクチャを使用することになされる。

複数ポートメモリシステムは各々のメモリモジュールと各々のプロセッサとの間に別々のバスを使用する。各々のプロセッサバスは各々のメモリモジュールに物理的に結合される。各々のメモリモジュールは複数のポートを有し、各々のポートはバスの一を受入れる。各メモリモジュールは所定の時間にどのポートがメモリをアクセスするかを決定する内部制御ロジックを有する。メモリアクセスに際して生じる衝突は各メモリポートに優先順位に割当てることによって解決される。プロセッサとメモリ間に複数のバスがあるため高い転送速度が実現される。マルチプロセッサクロスバススイッチアーキテクチャではプロセッサバスとメモリモジュールバスとの間の交点に切

換点が形成される。各切換点はプロセッサとメモリとの間に物理的な転送路を形成する制御ロジックを有する。制御ロジックはバス上を送られたアドレスを吟味し、その特定のモジュールがその時点でアドレスされた状態であるか否かを判定し、また同一のメモリモジュールに対する複数のアクセス要求を所定の優先順位に従って解決する。

マルチプロセッサタイムシェアアーキテクチャでは一定数のプロセッサが共通バスを介してメモリユニットに接続される。かかるシステムでは一度に一のプロセッサしかメモリと通信できない。転送動作は任意の時間にバスを制御できるプロセッサによって制御される。他の転送の開始を待っているプロセッサはまずバスの使用可能状況を判断し、バスが使用可能になって始めてメモリユニットをアドレスして転送を開始する。このシステムでは全てのプロセッサが一の共通バスを共用しているためメモリアクセスに際して衝突が発生する。メモリのとり

りにはアクセス要求のあるユニット間の優先順位を決定するバスコントローラによって解決される。このタイムシェアアーキテクチャでは一のプロセッサしかメモリと通信できず、他の全てのプロセッサは内部処理に使用中であるがバスがあくのを待っている状態になっており、従って不利である。

タイムシェア共通バスマルチプロセッサアーキテクチャよりも効率的なアーキテクチャはデュアルバスマルチプロセッサ構造であり、この場合一定数のローカルバスが各々ローカルメモリ及び一または複数プロセッサに接続されている。システムバスコントローラが各々のローカルバスと協働して各ローカルバスを共通システムバスへ結合する。多くの設計において、ローカルバスに接続される装置はローカルプロセッサによってのみ使用可能である。共通システムバスに接続されたメモリは全てのプロセッサで共用される。システムはローカルバスに接続された装置が他のローカルバスのプロセッサによってもアクセスできるように構成される。一つのプロセッサのみがシステムバスを介して任意の時間に共用メモリ及び他の共通をリソースと通信できる。ローカルバス上の他のプロセッサはローカルメモリ及びローカル装置との通信に使用中である状態に維持される。かかるシステムはマルチプロセッサシステムと考えてもよいが、より正確には複数コンピュータシステムとして分類されるものである。これはプロセッサ、メモリ、及び他の装置が互いにローカルバス上で接続された場合、かく形成されたローカルグループ自体が一つのコンピュータシステムをなすためである。

安全上の理由から多くのシステムは独立の動作する複数のコンピュータを有するリダンダントな構成にされている。他のリダンダントなシステムは分散プロセッサの半独立の動作を与えるもので、故障したプロセッサあるいはそれに関する装置がシステム全体の動作を破局的

に低下させることなくシステムから切離されるように構成されている。

分散マルチプロセッサシステムは最大の効率を得るためには半独立に動作する必要がある。相互接続信号リンクを有しつつ独立に近い動作を達成するためにはプロセッサオーバヘッドを回避する分散プロセッサの間の何らかのデータ転送手段を使用せねばならない。これはある程度までは各ポート毎に物理的に分離されたアドレス及びデータ入力バスを有する複数ポートメモリユニットを使用することで達成できる。プロセッサ間のハンドシェイクにより複数ポートメモリを媒介にしたプロセッサ間のデータ転送に必要な制御が得られる。しかし、共用メモリの使用はある程度のアドレス自由度を与えるものの、一度一つのプロセッサしかメモリをアクセスすることができない。この制約はいくつかのシステム設計上の要求、例えばプロセッサにメモリを自由にアクセスさせたいという要求と両立しない。

かかるシステム設計上の要求、すなわち共用メモリによって「待機」状態にされることをなくする問題の部分的な解決は先入れ先出し（FIFO）バッファをプロセッサ間に使用することで得られる。これによりプロセッサが直ちにメモリをアクセスするのが妨げられることがなくなり、データ流れの自由度が確保される。データは異なった速度で入出力でき、出力データは常にデータがバッファに入ったのと同じ順序になる。二方向のデータ流れについてFIFOバッファを両方向に使用することができる。しかし、共用メモリで得られるアドレッシングの自由度はFIFOバッファを用いた解決策では失われる。

上記の共用メモリ及びFIFOバッファを用いた解決策は個々の利点があるものの、いずれも別のメモリにより分離が与えられるにもかかわらず、最も望ましい処理の独立性の欠如により問題点を有している。共用メモリを用いた場合任意の時間にプロセッサをアクセスできないためシステム効率を低下させる望ましくない待機状態が発生する。FIFOバッファによりバッファリングされたCPUでは共用アドレッシングにより得られる情報が使用できない。すなわち、CPUはFIFOに書込む際データの位置割当てを制御することはできるがFIFOから読取る際のデータの割当てを制御できない。換言すれば、読取る際CPUは所望の記憶位置に到達するまでに関係のないデータを読取らねばならない。

より効率的な分散プロセッサの処理を行なうことがシステムが複雑になるにつれ重要になってくる。プロセッサの数が増加するにつれ相互通信条件はより複雑になり、望ましくない待機状態を生じる機械が増えてくる。そこで半独立動作を相対的に維持ないし増大させつつ処理効率を増大させる手段が必要になってくる。

本発明はプロセッサ間を結合するのに使われ、プロセッサ間のデータ転送を高効率で実行するモジュール式データハブとして機能する中枢の構成ブロックを提供する

ことを目的とする。

問題点を解決するための手段

本発明においては制御プロセッサないしマスタCPU及びこれと協働するメモリが同様な協働するメモリを有していてもよい別のプロセッサを含む一又は複数の分散した装置との間でデータの交換を制御プロセッサ及びこれを協働するメモリと他の各々の装置及びこれと協働するメモリ（これを有する場合）とをモジュール式データハブで結合することにより実行する。マスタCPUによるイニシャライズの後データハブはマスタCPUの命令により一又は複数の結合された装置へのデータ転送要求を生成する。一又は複数の要求が生成するとデータハブはマスタCPUによってデータソースとして指定された装置へのデータ転送要求の実行を開始する。ハブによる指定データソースへの転送要求の生成は直接メモリアクセス（DMA）サイクルを開始する。処理を維持するかハブの制御を放棄するかは決定権はデータソース側にある。データソースの命令処理は妨げられない。ハブは必要なデータをデータソースから得ることを記憶し、そのデータを指定された転送先のユニットにその転送先ユニットに好都合なタイミングで転送する。そこでハブは転送先ユニットに転送準備のできたデータを持っていて、転送先ユニットがデータを受取るのに好都合であると判断するまで待機中であることを知らせる。転送先ユニットがハブに転送に行なうように信号を送ると転送先ユニットは目下実行する必要がある無関係なプログラムの実行を再開し、一方ハブは転送先ユニットの協働するメモリを直接にアクセスすることが許される。転送先ユニットで使われているプロセッサによっては転送が終るまで通常のプログラムの実行を中断し待つ必要がある場合もある。ハブは複数の並列フォーマットデータチャンネルのみ、あるいは並列及び直列フォーマットチャンネルの組合せを含むことができる。

本発明ではまた各チャンネルはハブ内に形成されるプログラム可能優先順位を有し、これにより各チャンネルと協働する様々な装置からの競合するデータ転送要求を裁定することができる。そこでハブが一又は異なった優先順位を割当てられた転送要求を別々のチャンネルから受取った場合は最高の優先順位を有する装置がまず転送アクセスを許され、一方これと競合するチャンネルないしこの競合チャンネルの転送要求待ち行列に入れられて各々より高い優先順位の転送がプログラムされた転送優先順位に従って実行された後転送アクセスを許される。優先順位の構造はより高い優先順位の転送要求が生じた場合に実行中の低い優先順位の転送が中断されるようなものでもよい。この優先順位の低い転送は優先順位のより高い転送が実行された後に再開される。

本発明ではまた独立した割込条件の選択された組合せに応答するように再プログラム可能なマスク可能割込ラインがハブ中に形成される。

本発明ではさらに複数ポートデータハブが分散プロセッサシステムの構成における中核的な要素として使用される。複数のプロセッサ間をインターフェースするのにハブのようなモジュール式の建築ブロックに似た装置を使用することにより分散されたサブシステム間に望ましい高度な独立性を与える多様性のある構成要素が得られる。そこで複数のプロセッサをモジュール式複数ポートデータハブにインターフェースし、複数のモジュール式ポートデータハブを互いに結合することで多数のマルチプロセッササブシステムを有する複雑に分散したプロセッサシステムが構成される。各々のモジュール式複数ポートデータハブはデータ転送のいわば中心点として機能し、システム全体にかかるデータ転送の中心点が数多く形成される。

このように本発明によるモジュール式複数ポートデータハブは特に分散プロセッサシステムの構成に際し必要に応じていくらかでも複製できるデータハブを構成するモジュール式の建築ブロックにやや構成上の機能が似た装置であり、またマルチプロセッサ用途のより限定された目的にも使用できる。これは各チャンネルがDMA構造を有するマルチチャンネルインターフェース装置を用いてデュアルポートRAMなど共用バスのアドレッシングの独立性とFIFOアーキテクチャの情報流れの自由さとを結びつけるものである。データは分散プロセッサ間を極くわずかのプロセッサオーバーヘッドで転送される。

#### 実施例

本発明のこれらの、また他の目的、特徴及び利点は以下の図面を参照した実施例の詳細な説明より明らかとなる。

第1図は本発明による複数のモジュール式複数ポートデータハブ10を有する分散プロセッサシステムを示す図である。各々のハブは各々専用メモリを有するプロセッサ、I/Oコントローラ、I/O装置、インターフェース装置、及びその他多数の装置の一であってよい複数のデータソース／転送先ユニット12をインターフェースする。さらにデータソース／転送先ユニット12は例えば第1図に示したシステムと全く同一のシステム全体を含んでいてもよい。かかるリダンダントな構成はデータの一体性を確信するのに使用される。データリンク14は各々のデータソース／転送先ユニット12の協働するモジュール式複数ポートデータハブ10とを結合するのに使用される。かかるリンク14は直列でも並列でも、また同期的でも非同期的でもよい。

本発明による多数ポートデータハブ10はマルチプロセッサシステム、特に分散プロセッサシステム構成用のいわば建築ブロックに似たユニットとしての使用を意図している。しかし、本発明によるデータハブはまた複数プロセッサシステム中でも使用できることを念頭に置いておくべきである。設計者はデータハブ10をデータをソースと転送先との間でソース／転送先プロセッサオーバー

ッドをほとんど生じずに転送できる中枢的要素として使用することができる。

本発明によるハブ設計の本質的な特徴はデータソースとデータ転送先との間でメモリの共用を必要としない点にある。これにより従来のシステムで必要とされたオーバヘッドを除去することが可能になる。

第2図は各々2つのデータハブ26,28及び30,32を使用するように設計されている一対のリダンダントなシステム22,24を有するシステム20を示す。2つのシステム22,24は2つの別々の直列データリンク34,36を介して互いに通信することができる。そこで#1のデータハブ26はリンク34を介して#3のデータハブ30に直列に結合されている。同様に#2のデータハブ28は#4のデータハブ32に直列リンク36を介して直列に結合されている。一般にシステム22中の装置はどれも直列データリンク34,36のいずれかを介してシステム24中のあらゆる装置と通信することができる。しかし実際には指定されたシステム中の選択されたユニットのみが通常別のシステムの他の選択されたユニットと通信する。第2図のシステムアーキテクチャにおいては各々ハブ26,28,30及び32は並列リンク38,40,42,44,48,50,52,及び54を介してそれぞれ一対のローカルバス56及び58,56及び60,62及び64,及び64及び66をインターフェースする。

ローカルバス56,58,60,62,64,及び66は一群の装置68,70,72,74,76,及び78をインターフェースする。第2図の群の各々はCPUとメモリユニット(MU)とを含む。CPU及びMUの各々は協働するバスと同様に符号付けされている。換言すれば#1のデータハブ26は#1のローカルバス56及び#2のローカルバス58を#3のハブ30の結合する。かくて#16\*CPUはデータを#1のMUから#1のローカルバス56,並列リンク38及び#1のデータハブを経てシステム20中の他の装置のどれかに転送する。

同様に第3図にリダンダントのサブシステム82,84を有する別のリダンダントシステム80を示す。第2図のシステムと第3図のシステムの主要な差はハブ#1及び#2,またハブ#3及び#4の間にそれぞれ別の直列データリンク86,88が存在し、また同じサブシステム内でハブ間にローカルバスの共用がないことである。勿論第1,第2,及び第3図のシステムアーキテクチャは本発明によるモジュール式複数ポートデータハブを使用して構成される様々なアーキテクチャのわずかな例にすぎないことを理解すべきである。

第4図は本発明によるモジュール式マルチポートデータハブの概略的なブロック図である。第4図の複数ポートデータハブはデュアルポートデータハブであり、簡単のため2つのポートした有さない。しかし開示する本発明思想はデータハブ内に複数のポートが含まれる場合を包括するものであり、デュアルポートデータハブに限定されるものではないことを理解すべきである。

第4図の複数ポートデータハブ10はシステムA90とシ

ステムB92とをインターフェースするように示してある。システムA及びシステムBはそれぞれCPU95,96及び協働するメモリユニット98,100を有する。各々のシステムのCPUは概略的に単一のバスライン102,104として示したデータ、アドレス、及び制御バスを介して協働するメモリユニットと通信する。バス102及び104の各々は別々の複数ポートデータハブ10と別々のポートにおいて通信を行なう。このように、システムAとシステムBとの間のメモリの共用はされない。

システムAと複数ポートデータハブとの間にはハンドシェイク制御ライン106,108,及び110が設けられる。また同様にシステムBと複数ポートデータハブの間にはハンドシェイクライン112,114,及び116が設けられる。ハブと各々のCPUとの間の3つのハンドシェイク制御ラインはハブと各々のCPUを結ぶ2つのハンドシェイクラインにまで減少させることができることを理解すべきである。これは個々のCPUで必要なハンドシェイク思想如何による。

また第4図に概略化して示した複数ポートデータハブは上記のデュアルポートによる実施例に限定されることはなく、また並列インターフェースに限定されることはないことを理解すべきである。各々のハブは個々に形成でき、第5図に示すように多数の並列チャンネルと多数の直列チャンネルとを含ませるためのバス帯域幅によってのみ限定される。バス帯域幅はメモリシステムに可能な最大転送速度である。例えば100ナノ秒のアクセス時間を有するメモリ装置を有するメモリユニットシステムは10メガヘルツのバス帯域幅を有している。これは転送速度に対するチャンネルの総数を制御する。またシステム全体はそのシステム内の最も遅いメモリユニットの帯域幅で通信する。そこで第5図において複数ポートデータハブ10はNの並列チャンネルとZの直列チャンネルとの間でデータ転送をすることができる。換言すれば、本発明による複数ポートデータハブは様々なアーキテクチャにモジュール化することができる。

第6図は第4図と同様に2つのサブシステム120及び122とインターフェースする2つのポートのみを有する複数データハブ10を単純化して示す図であるが第6図は第4図よりやや詳細になっている。各々のサブシステム120及び122はそれぞれローカルバス124及び126を介して複数ポートデータハブ10とインターフェースされている。各サブシステムはそれぞれCPU128,130及びメモリユニット132,134を含む。

第6図の複数ポートデータハブ10は各々ローカルバス#1 124及びローカルバス#2 126と通信する2つのチャンネル136及び138及び138を有している。チャンネルA136はローカルバス#1 124からのライン140上の並列情報を受信して同じ情報をライン142を経てローカルバス#2 126に出力する。一般にライン140を経て受信されたデータはライン142へ供給されるデータと同じであるがア

11

ドレスは同じであったりなかったりする。チャンネルB 13  
8は#2のローカルバス126からデータをライン144を経  
て受信して同じデータを同一の、あるいは異なったアド  
レスでライン146を経て#1のローカルバス124へ出力す  
る。そこでチャンネルA及びチャンネルBは並列データ  
チャンネルの特徴を有する。しかしチャンネルAとチャ  
ンネルBとは本発明では単一の二方向チャンネルに結合  
することが可能なことを理解すべきである。

#1サブシステム120及び#2サブシステム122の転送  
速度に対する独立性を強化するため別のチャンネルが設  
けられる。

複数ポートデータハブのチャンネル136,138の各々は  
受信部148,150及び送信部152,154に分割される。これら  
の送受信部はワードカウントレジスタ156,158,160,162,  
バッファレジスタ164,166,168,170,及びアドレスレジスタ  
172,174,176,178を含む。

複数ポートデータハブ10はチャンネルAと協働するリ  
クエストフリップフロップ180及びチャンネルBと協働  
する同様なリクエストフリップフロップ182を含む。  
またメモリアドレスレジスタ184がチャンネルAと協働  
し、同様なメモリアドレスレジスタ186がチャンネルB  
と協働する。データハブ10はまだ制御ユニット188及び  
状態レジスタ190を含む。

各サブシステム120,122のCPUはアドレスレジスタ192,  
194,バッファレジスタ196,198,及びリクエストフリップ  
フロップ200,202を含む。各サブシステム120,122のメモ  
リユニット132,134はバッファレジスタ204,206,及びメ  
モリユニット208,210を含む。#1サブシステム120と#  
2サブシステム122との間のデータ転送は第7及び第8  
図のフローチャートを参照するとわかりやすい。

第7図は第6図の複数ポートデータハブ10内で同時に  
進行する2つのフローチャートの単純化した例である。  
スタートステップ200で動作を開始した後まず判定ステ  
ップ202で第6図のリクエストフリップフロップA180が  
1であるか否かが判定される。否であれば#1サブシス  
テムのローカルバス124をアクセスする要求がどのユニ  
ットからもなされていないことを意味する。従ってこの  
場合プログラムは204で示すステップ202から戻るループ  
を連続的に行ない、同一の質問をくりかえす。ローカル  
バス124に対する要求がなされている場合は次にステッ  
プ206が実行されて複数ポートデータハブ10がローカル  
バス#1 124のアクセスを要求する。次いでステップ208  
でアクセス要求がCPU#1 128によって許可されているか  
否かが判定される。否であればローカルバスのアクセス  
が許可されるまでプログラムはステップ208から210で示  
すループを循環する。アクセスが許可されると次いでス  
テップ212実行され、チャンネルA受信アドレスレジス  
タ172がその内容をメモリ#1アドレスレジスタ184へ転  
送する。チャンネルAの受信アドレスレジスタ172はCPU  
#1又はCPU#2のどちらか一方からそのどちらのCPUが

12

システム転送を制御しているか如何によってその内容を  
与えられる。次いでステップ214が実行され、アドレス  
されたロケーションのメモリスぺース1の内容がメモリ  
1のバッファレジスタに転送され、次いでチャンネルA  
の受信バッファレジスタ164に転送される。リクエスト  
フリップフロップAが次いでステップ216でゼロとさ  
れ、ステップ218でリクエストフリップフロップBが1  
とされ、またステップ220でCPU#1の一のバスが解放さ  
れる。プログラムは次いで222で示すループに達してス  
テップ202に戻る。

リクエストフリップフロップがステップ218で1にセ  
ットされるとその変化がステップ224で検知される。こ  
のステップ224での状態の1の変化の検知に先立ち第7  
図右側に示すプログラムは要求が検知されるまでステッ  
プ224をくりかえし循環している。検知の後、ステップ2  
26が次いで実行され、データハブ10がローカルバス#2  
126のアクセスを要求する。次いで判定ステップ228が実  
行され、CPUバスのアクセスが許可されているか未許可  
であるかが判定される。否であれば230で示す連続ループ  
がアクセスが許可されるまでくりかえされる。許可さ  
れると次いでステップ232が実行され、チャンネルAの  
送信アドレスレジスタ174の内容がメモリ#2のアドレ  
スレジスタ186に転送される。次のステップ234でチャ  
ンネルAの送信バッファレジスタ166に保持されているデ  
ータがメモリ#2のバッファレジスタ206の転送され、  
また次いでメモリスぺース#2 210の適当なアドレスへ  
転送される。送信バッファレジスタ166は制御ユニット1  
88に制御されて内容を受信バッファレジスタ164から受  
取る。同様な他のブロック間の転送が上記の転送がなさ  
れるに先立ち送受信部内でも生じている。次いでステッ  
プ236でリクエストフリップフロップBはゼロにセット  
される。転送ワードカウンタレジスタ158の内容は次い  
でステップ238で1減らされ、CPU#2のバスがステップ  
240で解放される。ステップ242で転送ワードカウントレ  
ジスタがゼロであるか1であるかの判定がなされる。否  
であればリクエストフリップフロップAはステップ244  
でゼロにセットされ、第7図左側のステップ202に後続  
するステップが全てのワードが転送されてしまうまでくり  
かえし実行される。

ステップ242のワードカウントレジスタがゼロである  
場合CPUはステップ246で転送が完了したと報告される。  
次いでステップ248が実行され、マスク可能割込がイネ  
ーブルされているか否かが判定される。割込がイネー  
ブルされている場合認定されたCPUがステップ250で割込ま  
れ、ステップ224への復帰がなされる。割込がイネー  
ブルされていない場合プログラムはCPUに割込むことなく  
直ちにステップ224に復帰する。勿論このフロープログラ  
ムは状態マシンを使用して実行できることを理解すべ  
きである。

第8図は単純化したフローチャートは典型的なサブシ

システム中のCPUにおける一連の事象及び複数ポートデータハブで生じている一連の事象を同時に示している。仮想的時間線が図の頂部から底部にかけて走っているものとする。この場合CPUは制御CPUであり、ステップ300でデータ転送用にデータハブ内にチャンネルが構成される。データハブはステップ302でチャンネルアクティビティを開始することにより応答する。一方CPUはステップ304に示す如く無関係なプログラムを自由に実行する。ステップ306でチャンネルが転送データを必要としていることあるいは転送すべきデータを有していることが判定されるとステップ308でリクエストフリップフロップAが1にセットされる。CPUはステップ310を周期的に実行して転送要求があるか否かを判定する。否であればステップ304で無関係なプログラムの実行に戻る。一方要求が有る場合はCPUはステップ312で自分自身が使用中であるか否かを判定する。使用中であればステップ304の無関係なプログラムの実行に戻る。否であればCPUはリクエストフリップフロップをステップ314で1にセットし、同時にそのプロセッサ構成如何によって転送が終了するまで待つか否を決定する。そのプロセッサ構成がCPUに転送が終了するまで待たせる必要がないようなものである場合CPUはステップ316で転送が終了するまでの待つことが不要であると判断し、次いでステップ318が実行され、CPUリクエストフリップフロップが再びゼロにセットされる。特定のプロセッサ構成のため関係のないプログラムの実行に進む前に転送が完了されるまで待つ必要がある場合はCPUはステップ316で転送終了まで待つことが必要であると判断し、次いでステップ320が実行され、転送が終了したか否かがくりかえし判定される。否であればステップ320がループ322で示すように転送が終了するまでくりかえし実行される。転送が終了すると次にステップ318が実行され、CPUリクエストフリップフロップがゼロにリセットされ、制御ステップ304へ再び戻され無関係なプログラムのCPUによる実行が継続する。

CPUリクエストフリップフロップがステップ314で1にセットされた後既にステップ324においてCPUリクエストフリップフロップがステップ314で1にセットされるのを持っているデータハブは次にステップ326を実行し、CPUとハブとの間でデータ転送がなされる。転送が終了した後データハブはステップ328でリクエストフリップフロップAを再びゼロにセットし、これによりステップ320でCPUは転送が終了したことを知る。

第9図は本発明によるモジュール式複数ポートデータハブ10の実施例を示す単純化したブロック系統図である。第9図の実施例は2つの直列入／出力ポート及び2つの並列入／出力ポートを有する。サブシステム#1は本実施例ではマスタCPUである。第9図のハブは例えば第3図のアーキテクチャに使用することができる。各々のハブは2つの直列データリンク及び2つの並列データリンクとインターフェースする。第9図の特定の実施例

は既に指摘したように第3図のアーキテクチャに使用するのが有利であるが、これは単に本発明によるモジュール式複数ポートデータハブ特定の実施例として選択可能な広範囲の内部構造に一にすぎない。第9図の実施例は従って本発明に教示を使用した特定のシステムアーキテクチャの単なる一例にすぎないことを理解すべきである。

第9図の実施例の2つの並列入／出力ポートの構造には第1の並列ポートの別個のデータバス350及び別個のアドレスバス352及び第2の並列ポートの多重化データ／アドレスバス354が含まれる。第2の並列ポートのライン354上のサブシステム#2へのハブ10からアドレス出力はデータがこの同じラインを通して送信したり受信したりされる前に外部でラッチする必要がある。

勿論、ここで開示する(第9図に関連して)ハブと衛星サブシステムあるいはハブ等を結合する特定のアドレスデータ、及び制御ラインの構造は単に図示した特定の実施例での使用状況により決まる便宜的なものにすぎない。従って本発明の範囲内で他の数多くのデータ転送結合構造が考えられ、ここで開示した実施例は本発明の範囲を限定するものではない。

図示の第9図に複数ポートデータハブ10は2つの別の複数ポートデータハブ102及び106とインターフェースされており、ハブ10はこれらと直列ライン360,362,364,366,1553遠隔端末インターフェース(RIT)368,及び第6図のサブシステムと同様な#1のサブシステム370及び#2のサブシステム372を介して通信する。サブシステム#1及び#2はどちらもCPUユニットとメモリユニット(図示せず)を含む。先に説明したようにサブシステム#1は本実施例ではマスタになっている。

第9図に示した本発明によるモジュール式複数ポートデータハブ10の実施例では6種類のデータ転送が処理される。転送は複数ポートデータハブ10及び#1サブシステム370,2つのマンチェスタ送信器(XMTR)/受信器(RCVR)対374,376,378,380を介して結合された2つのデータハブ102,106,1553RTI368,及びサブシステム#2122との間で生じる。図示の実施例は以下の優先順位のDMA転送が可能である。

- 1.1553RTIからサブシステム#1への、またサブシステム#1から1553RTIへの転送。
- 2.#3のハブ106からサブシステム#1のメモリへの転送。
- 3.#2のハブ102からサブシステム#1のメモリへの転送。
- 4.#3のハブ106へのサブシステム#1のメモリからの転送。
- 5.#2のハブ102へのサブシステム#1のメモリからの転送。
- 6.サブシステム#1のメモリからサブシステム#2のメモリへの、またサブシステム#2のメモリからサブシ

テム # 1 のメモリへの転送。

DMA要求は制御ブロック 382に含まれる制御ロジックと非同期である。各々の転送要求は上に示した順に優先順位が与えられ、番号 1 が最高優先順位をあらわし番号 6 が最低優先順位をあらわす。この 6 つの転送は各々今の場合マスタサブシステム # 1 内にあるデータ転送制御 (マスタ) CPUにより書かれるマスタワードによって個々にディスエーブル化される。16ビットアドレスバス 35 2及び354がサブシステム # 1, 1553RTI, 及びサブシステム # 2 のバスの双方に 64Kアドレススペースの全部を提供する。

マスタCPUはハブをこれがアドレスポインタ、ワードカウンタを与え、またデータ転送したいチャンネルをイネーブルするようにプログラムする。第15図は第9図のデータライン 350上を転送される様々な16ビットワードの定義をサブシステムチャンネル 1 からハブへアドレスライン 352上を送られる読取書込デコード命令と関連して示す。データラインの場合16ビットワードはハブ内の様々なレジスタに送られる。読取書込デコード命令の場合、読度デコード命令はライン 333上のチップ選択 (CSB) 信号と共にライン 331に生じるIOR信号によってハブへジクナルされる。また書込デコード命令はチップ選択命令と関連して生じるライン 335上のIOW信号によってシグナルされる。ハブをプログラムする場合はサブシステム # 1 がまず適当な書込デコード命令を用いてハブ内の適当なレジスタにアドレスポインタを書込む。次いでサブシステム # 1 は適当なワードカウンタレジスタを適当な書込デコード命令を再び使用してイニシャライズする。最後に書後デコード命令 9 (WR9) を使用して所望のチャンネルがサブシステム # 1 よりイネーブルされる。

#### 1553RTI転送の説明

転送の準備として1553RTI 368はライン 384を要求する。この要求 6 種類の転送の最高位の優先順位を有している。データハブ 10は以下の条件が全て満足されている場合この要求に応答する。

1. DMAチャンネル 1 がイネーブルされている。
2. その時点でのDMAサイクルが終了している。すなわちハブはこの時点で他のチャンネルのための動作をしていない。
3. サブシステム # 1 よりのMTC、すなわちライン 386上のメモリトグルクロック信号がローになっている。

第10図は1553RTI転送 (外部要求DMAサイクル) の典型的な波形のいくつかを示すタイミング図である。第10図 (a) には第9図中ライン 390上の12MHz信号を示す波形 388を示す。第10図 (b) はサブシステム # 1 のCPUのよりのライン 386上の信号を示すメモリトグルクロック (MTC) 信号波形 390を示す。MTCがハイの場合にDMA転送が許される。MTCがローの場合は個々のサブシステムのCPUが自己と協働するメモリユニットのアクセスを許され

る。MTCはサブシステム # 1 への要求に優先順位を付すのに使用され、これとバスのとりありを裁定するのに有効である。MTCはいくつかの実施例では図示のような規則的な再帰的信号であるが、そのパルス幅はプロセッサのクロック速度及び実行されているソフトウェア命令に従って変化してもよい (MTC「スローダウン」については第14図を参照)。データ転送と転送要求の内部での優先順位付けはいずれもサブシステム # 1 よりのMTCの立下り縁の位置によるのでMTCの変化はDMA転送に影響する。普通は第10図に示した特定の実施例ではMTCは乗算、除算、及びシフト命令群の間を除き周期的に脈動するこれらの命令群の間はMTCは仮りに複数のACK (確認) パルスが生じたとしても一様に保持されている。データハブはこの期間は最初の確認パルスのみを使用し他は無視する。従ってかかる命令が生じるとデータハブの有効転送速度が減少する。

第10図 (c) は転送に備えてRTI 368により発せられたライン 384上の要求信号を示している。第10図 (d) はハブ 10からサブシステム # 1 のライン 410上の要求信号 (REQ) を示す。第10図 (e) はライン 402上の信号を表わすバスイネーブル (BEN) 波形 400を示す。この波形はサブシステム # 1 がバスを差し迫ったデータ転送に備えてイネーブルすることを表わす。

同様に第10図 (f) はハブが差し迫ったデータ転送に備えてそのバス 350及び352をイネーブルすることを示す波形 404 (XOUTEN) を示している。転送が完了すると第10図 (g) に波形パルス 406で示した如き確認信号 (ACK) がライン 396に出力される。

#### マンチェスタ受信器の転送動作の説明

マンチェスタ受信器 374及び378の各々は協働する内部レジスタにおいてワード受信が終了するとワード転送を要求する。ワードは非同期的に受信され、また受信器は逐次的に受信されるワードを独立して受信しデータハブが必要とする並列フォーマットに組立てる手段を含む。2つの受信器 374及び378はそれぞれ優先順位レベル 2 及び 3 が与えられる。データハブは以下の場合にこれらの要求に応じる。

1. 各々のDMAチャンネル (2 又は 3) が書込デコード命令 0 及び 1 (第15図WRO及びWR1) によりイネーブルされている。これは適正なアドレスポインタが協働するレジスタ内に存在しておりチャンネルが書込デコード命令 9 (第15図WR9, D2ビット及びD3ビット (CH2又はCH3)) によってイネーブルされていることを意味する。
2. RCV×RDYフラグがアクティブ、すなわち受信期おRCV×RDYフラグがアクティブ、すなわち受信器が16ビットワードを完全にデコードしている。デコードが終了すると受信器は内部RCV×RDYフラグ (第15図D16ビット及びD15ビット) をセットアップし受信器がその内容 (R×2 RDY又はR×1 RDY) の転送準備を完了していることを示す。このフラグは状態レジスタ 408によって読取られ



る。

3.より高い優先順位の未定の要求がない。

4.この時点においてDMAサイクルが終了している。すなわち最後になされたDMAサイクルが終了している。

5. # 1 のサブシステム370のCPUからのMTCが立下った状態にある。

これらの要求が満足された後データハブ10により# 1 のサブシステム370のCPUに対しライン410上に要求信号

(REQ) が要求される。第11図(c)は要求信号REQの波形412と第11図(b)のMTC波形390及び第11図(a)の1 2MHz波形と関係を示す図である。ライン370上のMTC信号が立上りさらにライン402上のバスイネーブル信号(第1 1図(d) BEN, 波形414)がサブシステム# 1 の制御ロジックによって立上げられた後データハブ10はアドレスライン352上のアドレスポインタ、アドレス多重化器(ADD MUX) 416によりデータライン350上に形成される受信器データ、及び第11図(e)に波形420で示したライン418 10 上のメモリ読取(MRD)信号とを要求する。ライン396上の確認信号が転送が終了したことを示し、データバス350は確認信号(第11図(f) ACK, 波形424)の立上り縁で RCV×RDYフラグと共にディスエーブルされる。ライン410 20 上の要求信号(第11図(c) REQ, 波形412)は次いでライン386上のMTC信号が立下る際に除去される)。

マンチェスタ送信器の転送動作の説明

2つのマンチェスタ送信器(XMTR) 376及び380の各々は協働する送金器バッファが空である(すなわちTX×EMPフラグ(状態ワードレジスタの第13又は第14ビット)がアクティブな)場合、またそのチャンネルのワード転送カウントがゼロでない場合にはワード転送を要求する。図示の実施例ではマンチェスタ送信器が1ワードを 30 転送するのに20ミリ秒間を要する。送信器が使用中の場合はTX×EMPフラグ(送信器レジスタが空であることを示す)は偽となり他のワードがロードされる準備ができていないことを示す。このフラグは状態レジスタ(STATUS) 408で読まれる。2つの送信器376及び380にはそれぞれ優先順位レベル4及び5が割当てられる。複数ポートデータハブはかかる要求に以下の条件が満足されている場合応答する。

1.各々の複数ポートデータハブチャンネル(4又は5)がイネーブルされている。これはサブシステム# 7がハブがサブシステム# 1 のメモリから情報を検索するのに使用するアドレスポインタがセットアップされたことを意味する。アドレスポインタは書込デコード命令2又は3(第15図WR2, WR3参照)を要求するサブシステム# 1 によりセットアップされる。すなわちこれはアドレスライン352上に二進数の2又は3を与え、アドレスポインタをデータライン350上に設けてT×1又はT×2レジスタ381, 383に記憶することでなされる。これはまたサブシステム# が書込デコード命令3(WR3)を要求しア 40 ドレスラインが二進数の3を要求しデータラインがワー

ドカウントレジスタに入れられるカウント数を含んでいる(初めの8ビットに)ことをも意味している。最後にサブシステム# 1 は書込デコード9(WR9)を要求した二進数1をチャンネル4又は5のD4又はD5ビット(第15図参照)に設けることにより前記送信器をイネーブルする。

2.TX×EMPフラグがアクティブでTX×ワードカウントがゼロでない。

3.より優先順位の高い未決の要求がない。

4.ワード転送カウントがゼロでない。

5.現時点でのDMAサイクルが終了している。

6.サブシステム# 1 のCPUからMTCBが立下った状態にある。

これらの条件が満足されると複数ポートデータハブ10は第12図(c)の波形412で示するようなサブシステム# 1 のバスの要求信号(REQ)をライン410に要求する。第12図(aa)は第9図のライン390上に存在する信号に対応する12メガヘルツクロックパルス波形388を示す。第1 2図(bb)は第9図ライン386上のMTC信号に対応する波形3 90を示す。第12図(a)は第9図ライン402のBEN信号に対応する波形414を示す。第12図(b)はライン422上のMRW信号又はライン418上のMRD信号の一方に対応する波形416を示す。第12図(f)はライン396上のACK信号に対応する波形418を示す。

データハブによりサブシステム# 1 のバスのアクセス要求がなされ、またMTC信号が第12図(bb)に示すように立上った後ライン402上にバスイネーブル信号(BEN) (波形414)が要求される。イネーブル期間内にライン4 22上でのメモリ書込要求が実行された場合サブシステム# 1 のメモリはMRW信号波形416により示す如くメモリからハブにアドレスされたデータを書込む。確認信号波形432は第12図(d)に示すように要求されデータは送信器376又は380内にラッチされる。確認信号波形418が立 40 下るとワードカウントが減少しTX×EMPフラグがクリアされる。アドレスポインタは第12図(d)に示す如きライン396上の確認信号の立上り縁によって増やされる。ライン410上の要求信号(REQ)は次いでライン396上の確認信号(Ach)が開放された後の最初の12メガヘルツクロックによって除去される。

そこで送信器転送動作に際して生じる事象の順序を要約すると、送信器バッファ376又は380が空になり、すなわちTX×EMPフラグがアクティブになり送信器がより多くの情報を転送する準備ができたことが示されるとデータハブにより多くのデータをサブシステム# 1 のCPUへ転送する要求をなし、CPUはそのデータバスをMTCがハイである間イネーブルし、またサブシステムのアドレス/データラインを3状態化し、CPUはデータハウに確認を送ってDMAが発生したことを示し、さらに複数ポートデータハブはアドレスレジスタを津の転送のための新しい 50 アドレスの点まで増やす。この時点で送信器からハブ10

2,106の一へのデータの直列伝送の準備が完了する。

プロセッサ間のブロック転送

第9図の複数ポートデータハブ10は#1サブシステム370のプロセッサメモリと第2サブシステム382のプロセッサメモリとの間の二方向転送を支持する。この形の転送は最も優先順次が低く、以下の場合に実行される。

1. DMAチャンネル6がイネーブルされている。
2. より高優先順位の未決要求が存在しない。
3. ブロック転送ワードカウントがゼロでない。
4. 現時点でDMAサイクルが終了している。
5. 第9図のライン386上のMTC信号が立下っている。

転送の方向はBLK WRD CNT レジスタ429に記憶される内部ワードカウントワイヤの第9ビット（第15図の書込デコード命令8（BLK WRD CNT）を参照）により決定される。第9ビットがハイであることはサブシステム#2側の読取サイクル及びサブシステム#1側の書込サイクルを示す。同様にこのビットがローであることはサブシステム#1のバスの読取サイクルとそれに続くサブシステム#2のバスの書込サイクルとを示す。

DMAチャンネル6がイネーブルされるという仮定はサブシステム#1のブロックアドレスポインタ（第15図書込デコード命令6を参照）が既にサブシステム#1によりイニシャライズされていることを意味する。そこでサブシステム#1のブロックアドレスポインタはサブシステム#1からサブシステム#2への転送の際サブシステム#1からのデータが得られるアドレス、あるいはサブシステム#2からサブシステム#1への転送の際データが記憶されるサブシステム#1中のアドレスを含んでいる。同様にサブシステム#2のブロックアドレスポインタ（第15図書込デコード命令7（WR7）を参照）もまたサブシステム間でのデータ転送の際情報が取出されたり入れられたりするアドレスを内部に有している。ブロックワードカウント（BLK WRD CNT）レジスタはまた第15図の書込デコード命令9（ワードをマスクイネーブル）におけるビット6のイネーブルによってイネーブルされる際イニシャライズされる。

第9ビットのワードカウントがロー、すなわち第9図の#1サブシステム370から#2のサブシステム372へのデータ転送が指令されると複数ポートデータハブ10はサブシステム#1から送信器への転送に関連して説明したような#1のサブシステム370との間での読取りDMAサイクルを開始するが、その際データハブに入室するデータが送信器レジスタ376,380でなくATEMPレジスタ430に記憶される点が異なる。データはライン396上の確認信号（第12図（d）の波形432を参照）の立上り縁を使ってレジスタ430内にラッチされる。ライン396上の確認信号が立下るとライン434上にバス要求信号BR（第12図

（c）波形434を参照）が制御ユニット382によって要求される。サブシステム#2のCPUが空くと、すなわちライン438上のバス許可信号BG（第12図（f）波形439を参

照）がハイになるとライン440にバス許可確認信号（BGACK）がハブにより要求され（第12図（g）波形442を参照）先行のバスサイクルが終了してハブガバスのマスタになっていることが示される。

次いで転送先のアドレスが多重化アドレス/データバス上に要求され（第12図（k）ADBUS、波形446参照）、安定なアドレスが存在するとハブがライン448にCASLE信号（第12図（h）波形449を参照）を要求する。これはラッチ450中のアドレス情報をラッチしてサブシステム#2へのアドレスライン452に供給される。アドレスはBINレジスタ456からアドレスを与えられる多重化器（MUX）454により与えられる。BINレジスタ456はプログラムされたアドレス情報をデータライン350を介してサブシステム#1より与えられる。

データは次いでライン460上にLUDS信号第12図（i）波形462を参照）を制御ユニット382によって要求することでサブシステム#2のメモリに書込まれる。このサイクルはサブシステム#2によりライン466上に要求される。第12図（j）に波形464で示した確認信号DTACKにより終了する。サイクルの終りにデータ確認信号がまず要求されることはまずライン388上の12メガヘルツクロック信号と同期される。これは次いでサブシステム#2の全てのラインを3状態化して要求を除去し、サブシステム#2のサイクルの終りを示すのに使われる。

ブロックワードカウントの第9ビットが1である場合複数ポートデータハブはまずサブシステム#2の側の読取サイクルを実行し、これをデータラッチ（BTEMP）470内に一時的に内部ラッチし、次にサブシステム#1の側の書込サイクルを実行する。

上記の場合の全てにおいて、第12図（d）又は（j）に示したようなライン396又はライン466上の確認信号の立下り縁はブロックワードカウントを減少させる。また確認信号の立下り縁はそれぞれのアドレスポインタを増やす。ブロックがサブシステム#1のメモリからサブシステム#2のメモリへ動く場合、確認信号は第9図をライン466上の同期した同期確認信号になる。ライン472上のイネーブル信号（DMAEN）が否定されるとその時に優先されていた要求はクリアされ（またライン410上の要求信号もクリアされる、第12図（c）参照）、それ以上の要求が生じるのを抑止する。これは基本的な要求には全く影響しない。そこでライン472上にイネーブル信号が再び出力されるとMTCの立下り縁が再び復元順位を与え、必要に応じて新たな要求を発生させる。

第1図を参照するに、第9図の#2サブシステム372から#1のサブシステム370へデータ転送するのに際関係するタイミング関係のいくつかが示されている。第13図（a）の波形388は第9図中ライン390上の12メガヘルツ信号を示している。ライン386上のMTC信号は第13図（b）に波形390として示してある。ライン434上のハブからサブシステム#2へのバス要求信号（BR）が第13図

(c)に波形480として示してある。ライン438及び440上のバス許可(BG)及び確認(BGACK)信号はそれぞれ第13図(d)及び(e)の波形482及び484として示してある。サブシステム#2からサブシステム#1へのブロック移動の際の手順は次の通りである。まずデータハブ10がサブシステム#2に対しライン434上にバス要求信号を出力する。サブシステム#2は空くとライン438上に第14図(c)及び(d)の波形480及び482のようなバス許可信号を出力する。データハブは次いで440上にバス許可確認信号GBACK(第13図(e)の波形484を参照)をライン440に出力し、ライン434上のバス要求信号をゲートオフし、サブシステム#2の読取/書込ライン447をローに駆動し(第13図(i)波形490を参照)、転送先アドレスを多重化アドレス/データバス354(第13図(h)波形491を参照)に要求する。アドレスが安定するとデータハブはライン448にASLE信号を出力し(ハブからのハブ/ラッチイネーブル出力へのアドレスストローブ入力;第13図(f)波形493を参照)、これはデータラッチ450からのアドレスを反多重化する。サブシステム#2内のストローブされたアドレス中にストアされたデータはデータハブが読取/書込信号(CPUWR)をライン447上に出力することによりADバス354を経て一時記憶レジスタ(BTEMP)470へデータ流れ及びライン460上のLUDS信号をハブ内へ導く目的で送られる。第13図(g)及び(i)の波形488及び490をそれぞれ参照のこと。

そこでデータハブはデータをサブシステム#2からライン354へ経て記憶レジスタ(BTEMP)470へライン460上のデータストローブ信号(LUPS)を用いて受入れる。サブシステム#2のサイクルの終りは第13図(j)のDTACK信号波形492によって示される。するとデータハブはその制御ユニット382への内部バス要求信号を発生させる。MTC信号はバス要求に際してより高い優先順位の未決の要求が存在していなければサブシステム#1への要求信号をラッチしてライン410に出力する。これの第13図(k)に波形500で示す。一方サブシステム#1は第13図(l)に波形502で示すバスイネーブル信号をライン402に出力し、データ転送を開始する。これはサブシステム#1をアドレス/データライン352,350に対して再び3状態化する。次いでレジスタ470から多重化器386及びデータライン350を経てサブシステム#1へ到るデータ転送が生じる。これを第13図(m)に波形504で、また第13図(n)に波形506で示す。転送はライン402上のバスイネーブル信号を受信した後データハブによって開始される。転送が終了するとサブシステム#1はDMAが発生し、データがサブシステム#1メモリに書込まれたことをライン396に確認信号を発生することで示す(確認信号については第13図(n)を参照)。双方のアドレスポインタは次いで増加され、ワードカウントは減少される。ワードカウントがゼロでないとゼロなるまで全過

程がくりかえされる。

#### DMA転送のためのタイミングの関係

第14図はサブシステム#1からサブシステム#2への1ワードブロックの転送のタイミングを示す。 $T_1 \sim T_4$ はデータハブと2つのプロセッサとの間に同期のため変化する。サブシステム#1の側の転送はMTC信号とタイミングが整合している。サブシステム#2の側の転送は12MHz信号と整合している。第13図のサブシステム#2の側の転送はハブとサブシステム#2との間の信号がMTC信号と整合していることを示しているが、これは必ずしも必要ではない。例えば第14図ではMTC信号とハブとサブシステム#2との間の信号との間には何の関係も存在しない。

ここで導かれた $T_1 \sim T_4$ はサブシステム#2においてモトローラMC68000とインターフェースするハブについてのものである。しかし他のプロセッサを使用した場合タイミングが変化する。

$T_1$ はハブがバス要求信号をサブシステム#2へ送りサブシステム#2からバス許可確認信号が返るまでの時間である。MC68000の仕様に基き $T_1$ の最小時間は制御CPUのランニング周波数の同期の1.5倍に選ばれる。図の例ではCPU2は8メガヘルツで動作している。同様に $T_1$ の最大時間は制御CPUのランニング周期の4倍であり、上の側では500ナノ秒である。この時間はCPU2の命令サイクルのどこでバス要求がなされるかによって多重化する。要求がサイクルの途中であるとデータハブはアクセスする前にASLEが否定されるまで待たねばならない。第14図

(e)はデータハブから要求している装置でのバス要求信号である波形700、及び第14図(g)の波形702で示すバス確認信号の立下り縁によりこのバス要求信号が否定されるまでの時間を示している。

バス許可確認信号440がハイになってからライン448上の信号ASLEの立下りによって示される有効なアドレス、すなわちデータからの反多重化アドレスが現われるまでの間の時間 $T_2$ は少なくとも83ナノ秒(12MHzの1サイクル分)で最大値は167ナノ秒に達する。この変化は確認信号と12メガヘルツクロック信号との内部周期によるものである。すなわちライン440上のBGACK信号(第14図(g)波形702)は12MHz信号と非同期であり、12MHzの1サイクル分の時間窓(83ナノ秒)内の任意のところで生じる。そこで第14図(d)に波形704として示す12メガヘルツ信号は第14図(h)の波形706で示すAKLE信号の立下りをトリガする。波形702で示すバス許可確認信号の立上り縁からBGACK信号の後波形704で示す12メガヘルツクロック信号の第2番目の立上り縁によってトリガされるASLE信号の立下り縁までの時間 $T_2$ は上記の如く最小83ナノ秒、最大167ナノ秒になる。

第14図(i)に波形708で示す第9図ライン460上のLUDS信号の接続時間 $T_3$ は最小83ナノ秒、最大値は83ナノ秒の整数倍になる。この時間はLUDS信号と第14図(i)に

波形710で示すDTACK信号466との同期によって変化する。最小時間はライン378上のDTACK信号を接地することで得られる。第14図(i)に示すようにLUDS波形708の立下り縁はアドレスが外部ラッチ450へのストローブを終った後最初に現われる12メガヘルツクロックパルスの立上り縁に対応して形成される。

要求相互の内部優先順位の割当ては第14図(b)に波形712で $T_4$ で示す時間内に行なわれる。内部優先順位割当て時間( $T_4$ )はLUDSの否定の後、すなわちデータがストローブされた後始まり次の要求がCPU1よりなされるまで続く。

要求に内部優先順位を割当てするのに必要な時間は他の未決の要求、制御プロセッサのスピード、及びMTCの立下り縁の出現によって変化し、0から666ナノ秒の間である。

転送時間は $T_1 + T_2 + T_3 + T_4 + 74$ ナノ秒 + 666 + 83ナノ秒である。最悪の場合加算転送時間は(通常332ナノ秒の $T_3$ を含めて)2,454ナノ秒である。また最小の転送時間は1,351ナノ秒である。データハブをプログラムする場合アドレスポインタがまず書かれる。次いでワードカウントレジスタがイニシャライズされる。最後に所望のチャンネルがイネーブルされる。

2つのマンチェスタ受信器のDMAチャンネルは本実施例では増加していく16ビットアドレスポインタカウンタを有する。ワードカウントレジスタを使用することはできない。しかし勿論別の実施例でワードカウントレジスタを使用することは可能である。従って上記実施例においては受信器は受信器チャンネルがディスエーブルされるまで連続的にデータを隣接するロケーションに記憶しつづけることになる。

本発明においてはDMAを介して受信器により転送されるワード数を制御するのにロックステップモードが有用である(第15図の書込デコード9及び読取デコード4のビット7及び8を参照)。状態ワード(第15図読取デコード2を参照)において受信器チャンネルの状態を決定するのに受信器当り3つのフラグが使用可能である。RX1及びRX2RDYはワードがデコードされていて転送準備ができていることを示す。この信号は優先順位ロジックに対する要求として作用する。RX1BSY及びRX2BSYは受信器Xが目下ワードデコードのために使用中であることを示す。この信号はいつ受信器のDMAチャンネルをディスエーブルするか、あるいはいつアドレスポインタを変化させるかを決定するのに有用である。RX1 BKL VLD及びRX2 BKL VLDはそれまでに受信した全てのデータがバリティーもマンチェスタエラーも有していないとき真になる。このビットはCPU1が受信器のパッファを読取ると、あるいは新しい受信器アドレスポインタが書かれるとクリアされる。

このように本実施例に設けられたロックステップモードはリダンダントなチャンネルとやや同期した状態に維

持する手段を提供する(第15図書込デコード9のビット6及び7を参照)。このモードでは送信器と受信器のDMAサイクルは相互に織りこめられている。そこでロード送信器DMAサイクルの後にはRX×RDYフラグに依存しTX×EMP及びXMTR×WC>0フラグとは独立な記憶受信器DMAサイクルが続く。同様に記憶受信器サイクルの後にはTX×EMP及びXMTR×WC>0フラグに依存しRCV×RDYフラグと独立なロード送信器DMAサイクルが続く、各々の送信器ワードカウントレジスタをロードすると送信器DMAチャンネルはイネーブルされ受信器の要求はブロックされる。リンク対1及び2を書込デコードを介してロックステップモードに独立に加えてもよい。ロックステップイネーブルビットはイネーブル読取において読取られる。

リンク対はまた自己検査ラップアラウンドモードで独立に加えてもよい。このモードでは送信器出力は受信器入力へ向けられ外部マンチェスタの出力はディスエーブルされる。ラップアラウンドモードイネーブルビットはイネーブル読取において読取らる。システムは電源投入によって自動的に自己検索モードにはならない。

20 マスク可能割込み

各データハブは第9図に示す開いた外部ドレイン割込線720(INTR)を有する。このラインは以下の8つの割込条件のどの組合わせにも応答するようにプログラムされる。

1. TX1パッファが空の場合に割込む。
2. TX2パッファが空の場合に割込む。
3. RCV1でワードが準備完了している場合割込む。
4. RCV1でワードが準備完了している場合割込む。
5. TX1 DMA ブロック転送がなされた場合に割込む。
- 30 6. TX2 DMA ブロック転送がなされた場合に割込む。
7. DMAブロック移動がなされた場合に割込む。
8. メールボックスデータが得られる場合に割込む。

これら全ての条件は書込デコード命令9の上位バイト(第15図参照)によりマスク可能である。マスクレジスタ及び全ての割込条件は状態読取(第15図読取デコード4を参照)において読取可能である。

16ビットワードがサブシステム#2によりDMA1にメールボックスレジスタ(MAIL)721に書込まれる。このデータはライン447上のCPURW及びライン722上のCTRL信号がローである場合LUDS8の立上り縁でメールボックスレジスタ内の配送される。書込まれた際MBX DTA AVLBフラグが読取デコード2のビット1にセットされメールボックスデータが使用可能なことを示す。このビットはサブシステム1がこのメールボックスデータを多重化器386及びデータバス350を介して読取るとクリアされる。

サブシステム#1はまたメールをサブシステム#2へ送る。デコードユニット724は書込デコード命令9のビット16をデコードして内部11ビットラッチへ書込む。このラッチはサブシステム#2バス上で読取可能である。読取デコード命令の上位6ビットはハブ内の状態情報よ

25

りなる。サブシステム#1へ書込まれるとMXB DTA AVL Bフラグが読取デコード2のビット8にセットされる(第15図状態ワード読取を参照)。このフラグはサブシステム#2が多重書器454を介してメールボックスを読む際にクリアされる。

またサブシステム#1がサブシステム#2の64Kアドレススペースの任意の場所のデータワードを間接的にDMAできるオプションを設けてもよい。16ビットデータワードをサブシステム#2へ送るにはサブシステム#1はまずサブシステム#2のポインタ(第15図書込デコード命令に参照)をセットアップして一のワードの転送先アドレスを指定する。サブシステム#1は次いでデータをハブに書込み、ハブはサブシステム#2へのデータのDMA転送を開始する。これはブロック転送DMAチャンネルがイネーブルされている間は使えない。

なお本発明の技術思想の範囲内で様々な変形、変更が可能である。

#### 発明の効果

本発明は信号プロセッサ間のデータ転送を高い効率で実行するデータハブを提供する。データバスはマスタCPUから低い優先順位のデータ転送命令を受け、またかかる低い優先順位のデータをプロセッサ間でデータソースあるいは転送先に著しいオーバヘッド負担を課することなく転送することを可能にする。ハブはさらに中位の優先順位のデータ転送を非同期的に受取り、受取ったデータを記憶しまたこれを他の低い優先順位の転送がなされる前に転送先へ転送することが可能である。ハブはさらに中位及び低い優先順位の転送が実行される前にハブによって実行される高い優先順位の転送要求を非同期的に受けることができる。このハブは信号プロセッサを結合する中枢的要素として使われる。

#### 【図面の簡単な説明】

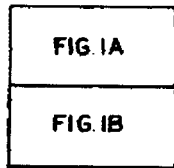
第1A図及び第1B図は夫々中枢的構成要素として使われる本発明による複数のモジュール式複数ポートデータハブを有する分散プロセッサシステムを示す図、第2図は各々2つのデータハブを使用するように設計された一対のリダンダントなシステムを示す図、第3図は各サブシステム内に2つのデータハブを使用しているリダンダントなシステムを示す図、第4図は本発明によるモジュール式複数ポートデータハブを示す概略的ブロック図、第5図は複数の並列及び直列チャンネル間でデータ転送を行なえる本発明による複数ポートデータハブを示す概略的ブロック図、第6A図及び第6B図は夫々2つのサブシステムをインターフェースする2つのポートのみを有する本発明による複数ポートデータハブの概略的ブロック図、第7A図及び第7B図は夫々第6図の複数ポートデータハブ

26

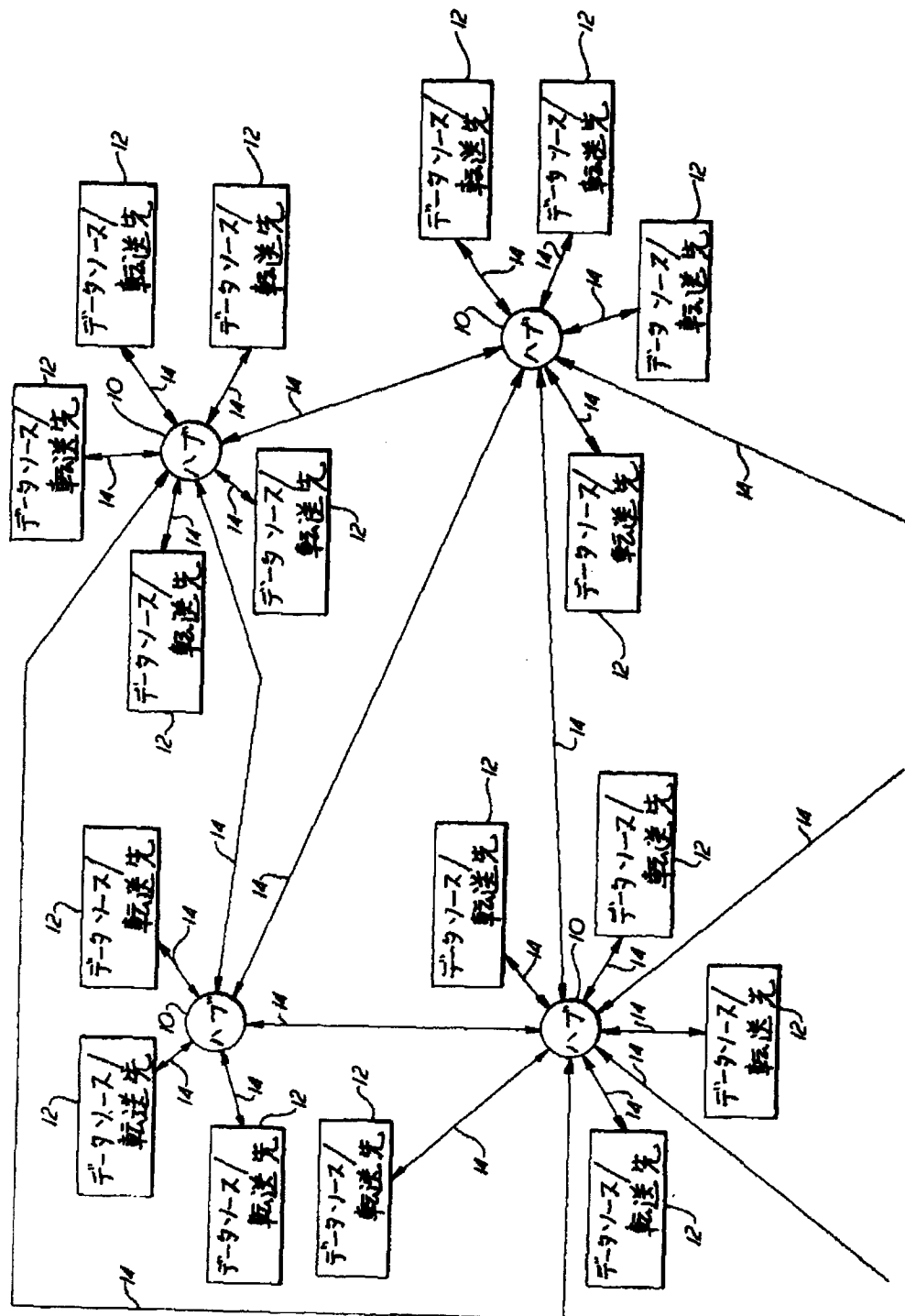
内で実行されるステップを示す2つのフローチャート、第8図は典型的なサブシステムCPU内で実行されるステップと本発明による複数ポートデータハブで実行されるステップとを同時に示すフローチャート、第9A図及び第9B図は夫々本発明によるモジュール式複数ポートデータハブの一実施例を示す概略的ブロック図、第10図は第9図のハブを使った直列伝送の典型的な波形を示すタイミング図、第11図はマンチェスタ送信器の転送の際第9図のデータハブ内で生じる一連の事象のタイミング図、第12図は第9図のサブシステム#1及び#2内のCPU間でデータ転送を行なう際のタイミング関係を1ワードブロック転送の場合について示す波形図、第13図は第9図サブシステム#1からサブシステム#2へのデータ転送に参与する信号のいくつかのタイミング関係を示す図、第14図は第9図サブシステム#1からサブシステム#2への1ワードブロック転送のタイミングを示す図、また第15A及び第15B図はそれぞれ第9図のデータハブはワードの定義を下位及び上位バイトについて示す図である。

10,26,28,30,32,102,106……データハブ、12……データソース/転送ユニット、14,34,36,38,40,42,44,48,50,52,86,88……リンク、20,22,24,80,82,84,90,92,120,122,370,372……システム、56,58,60,62,64,66,102,104,124,126,350,352,354……バス、68,70,72,74,76,78……装置、94,96,128,130……CPU、98,100,132,134,208,210……メモリユニット、106,108,110,112,114,116,140,142,144,146,333,335,350,352,360,362,364,366,370,386,388,390,397,402,410,418,422,434,438,440,447,448,452,460,466,722……ライン、136,138……チャンネル、148,150……受信部、152,154……送信部、156,158,160,162……ワードカウントレジスタ、164,166,168,170,196,198,204,206……バッファレジスタ、172,174,176,178,192,194……アドレスレジスタ、180,182,200,202……フリップフロップ、184,186……メモリアドレスレジスタ、188,382……制御ユニット、190,408……状態レジスタ、200,202,206,208,212,214,216,218,220,224,226,228,232,236,238,240,242,244,246,248,250,300,302,304,306,308,310,312,314,316,318,320,326,328……ステップ、204,210,230,322……ループ、210……メモリスペース、368……RTI、374,378……送信器、376,380……受信器、381,382,430,456,470,721……レジスタ、386,416,454……多重化器、388,390,400,406,412,414,416,418,420,424,432,433,437,442,446,448,464,480,482,484,490,491,493,500,502,504,506,700,702,704,706,708,710,712……波形、450,472……ラッチ、466……データ確認信号、724……デコードユニット。

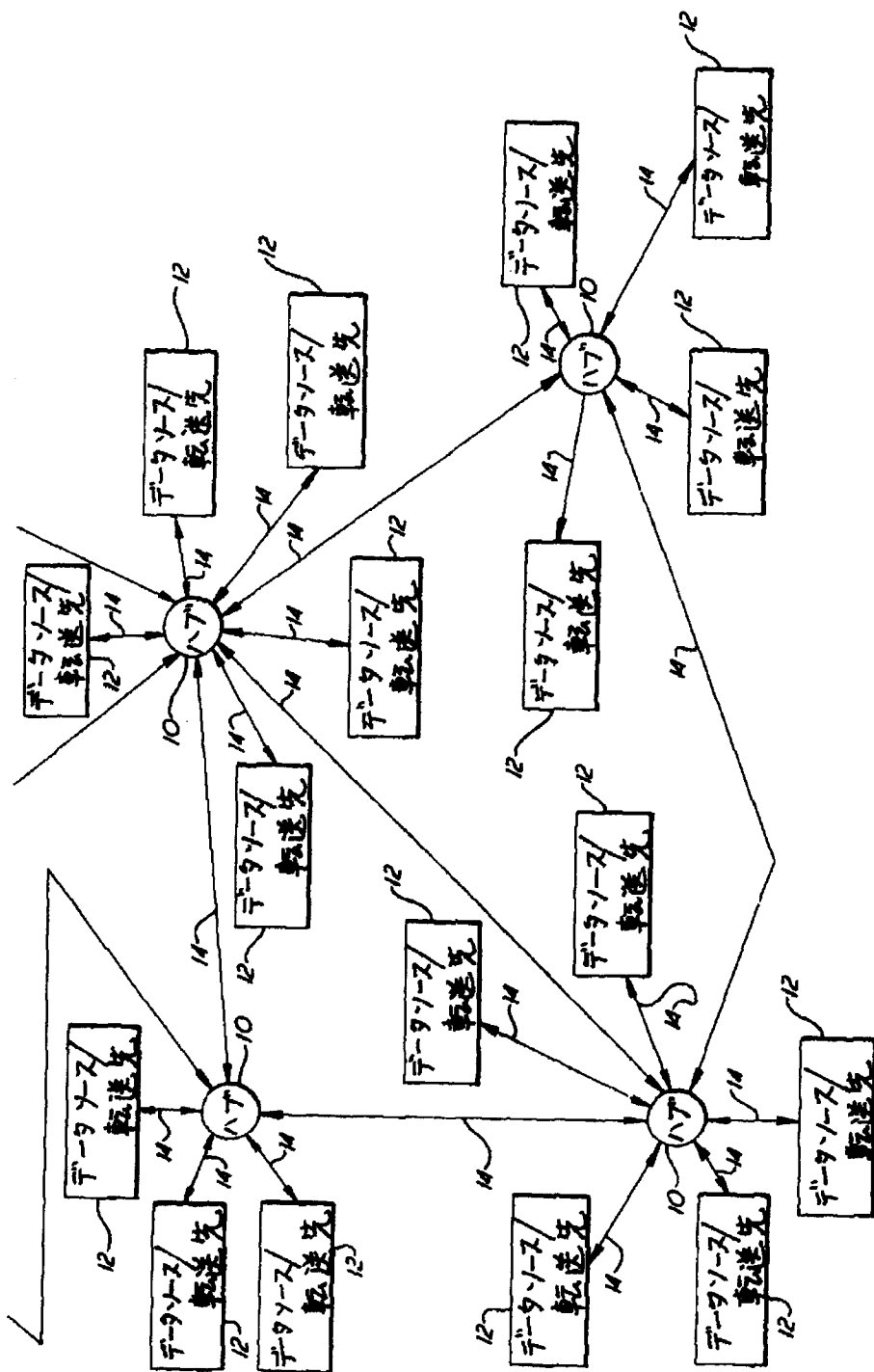
【第1図】



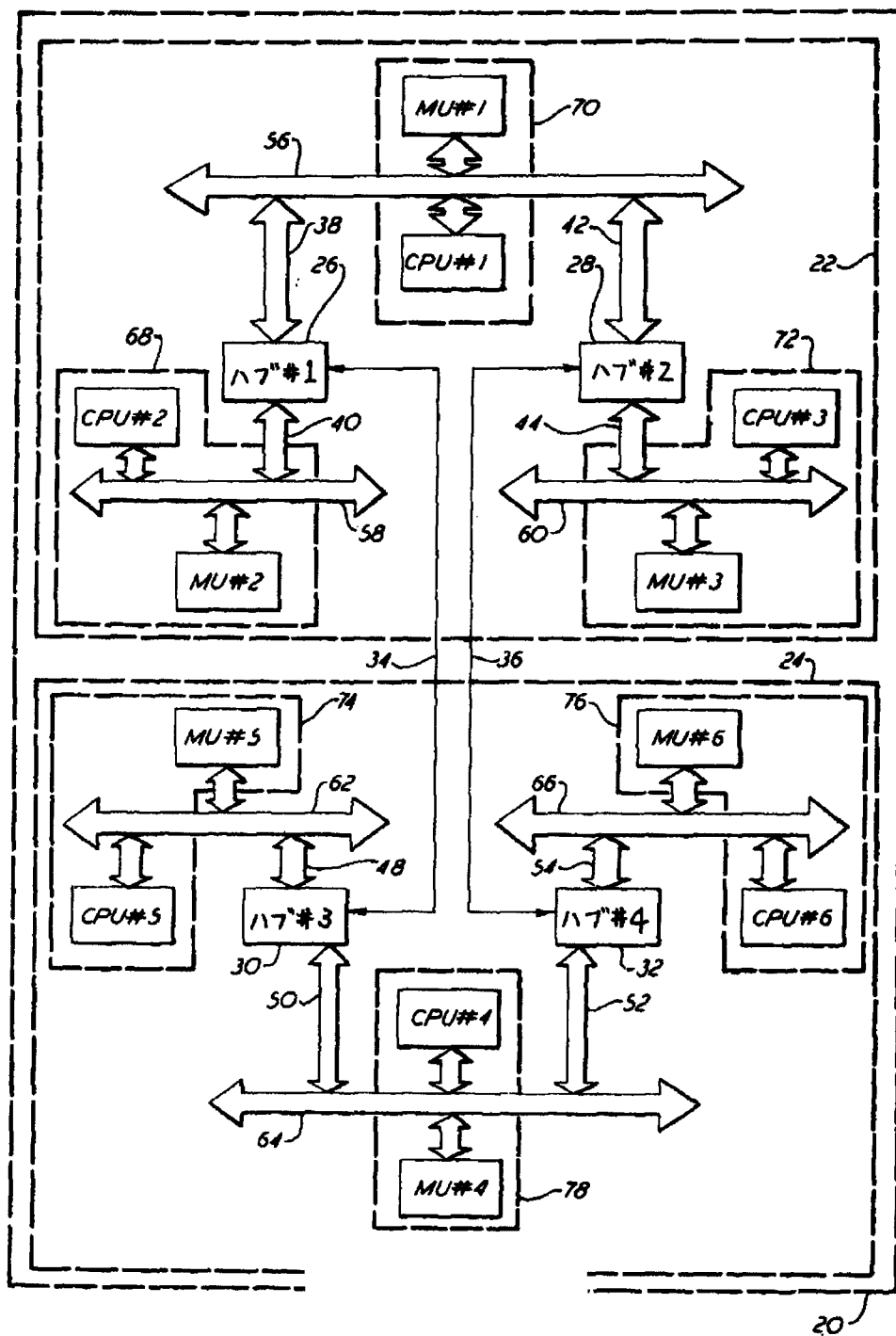
【第1A図】



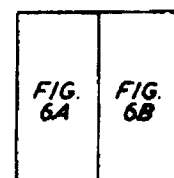
【第1B図】



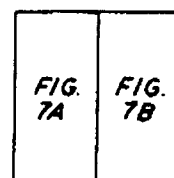
【第2図】



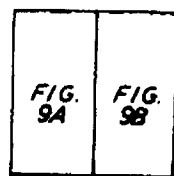
【第6図】



【第7図】

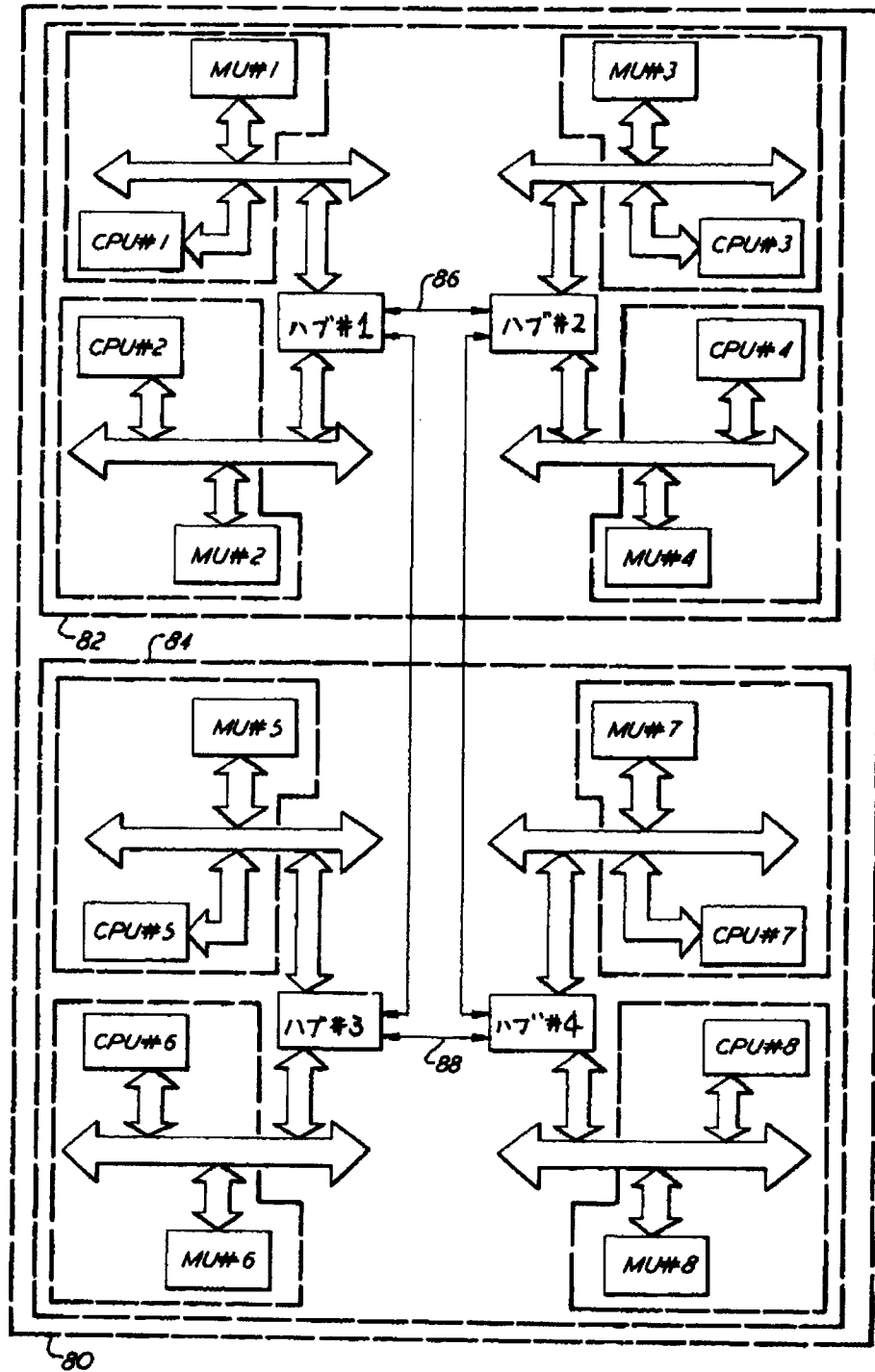


【第9図】

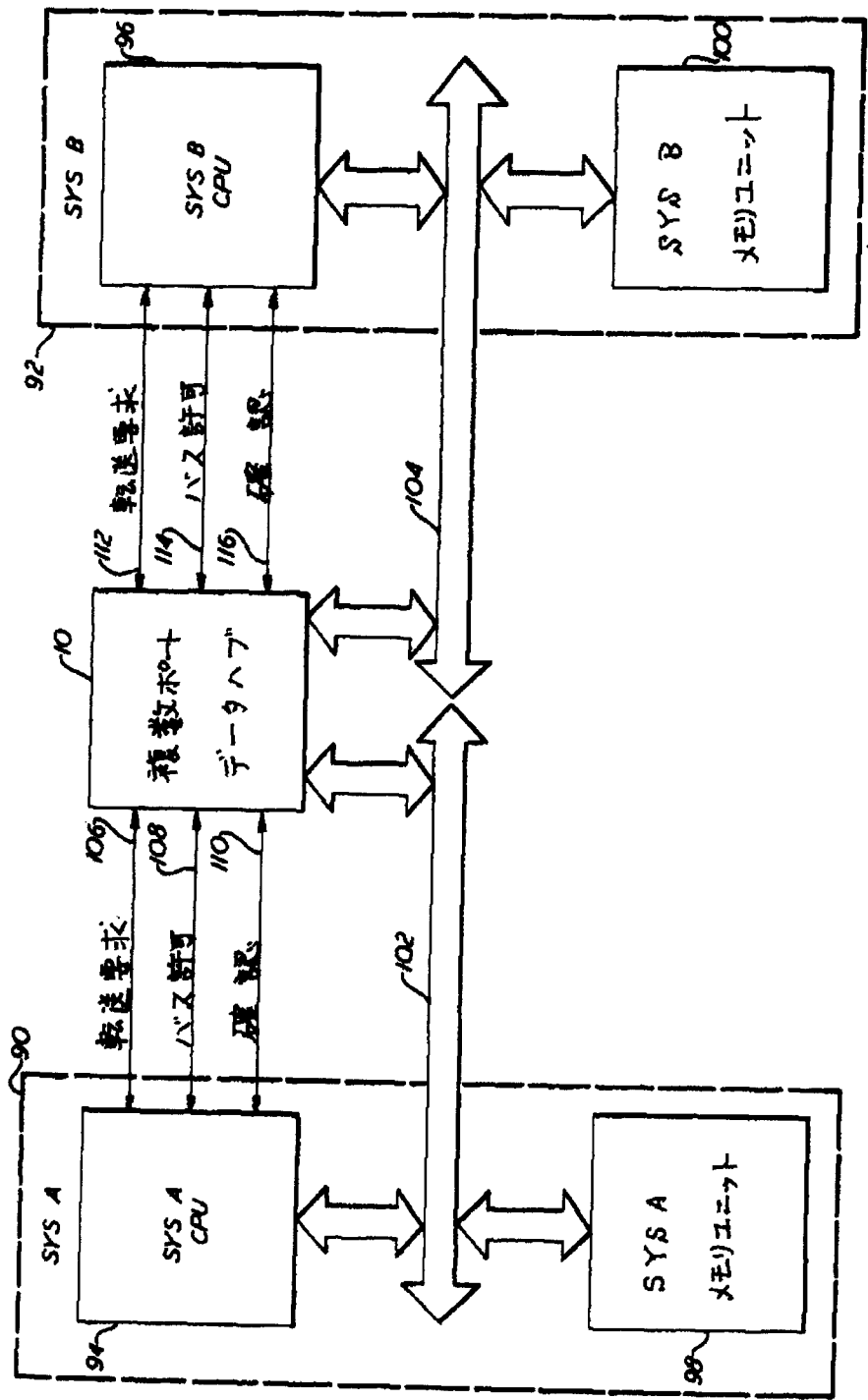




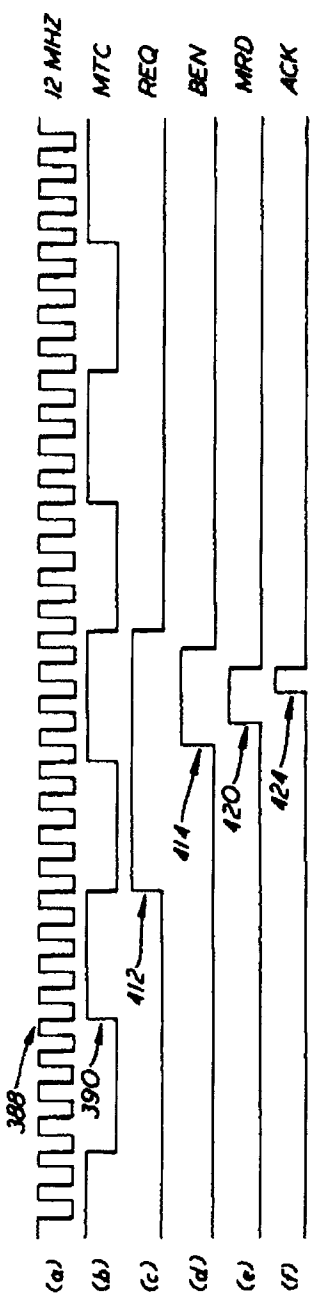
【第3図】



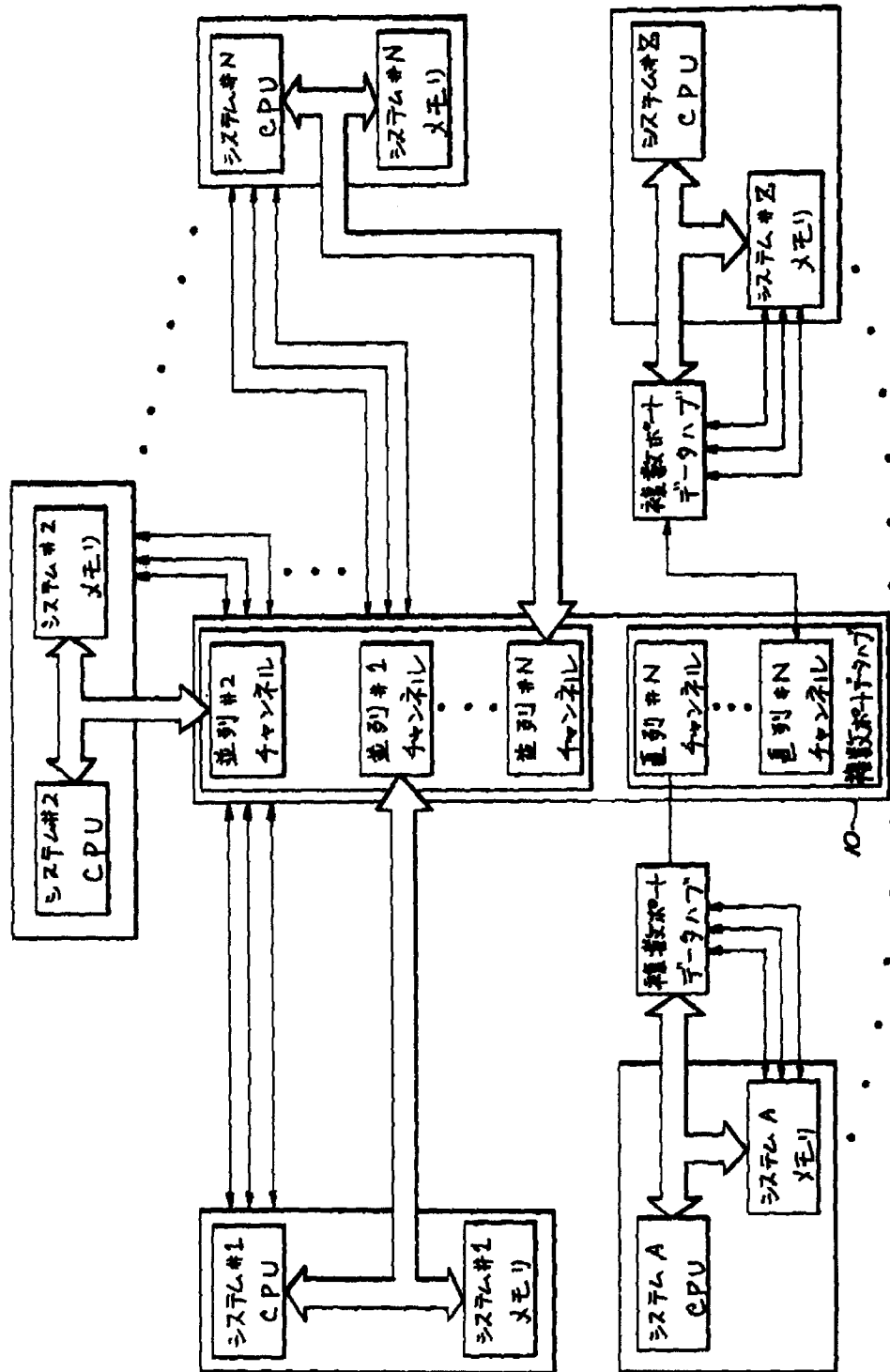
【第4図】



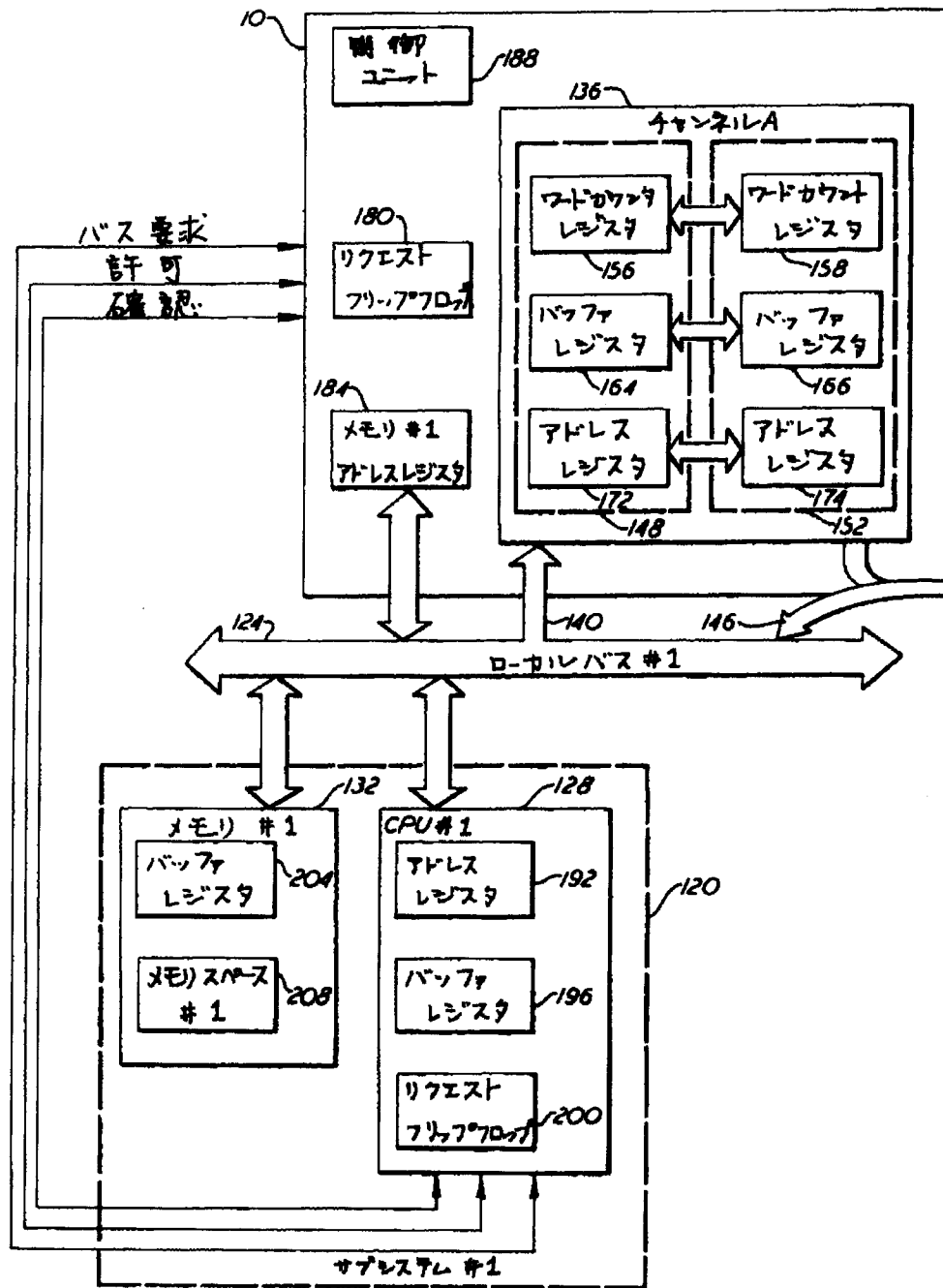
【第11図】



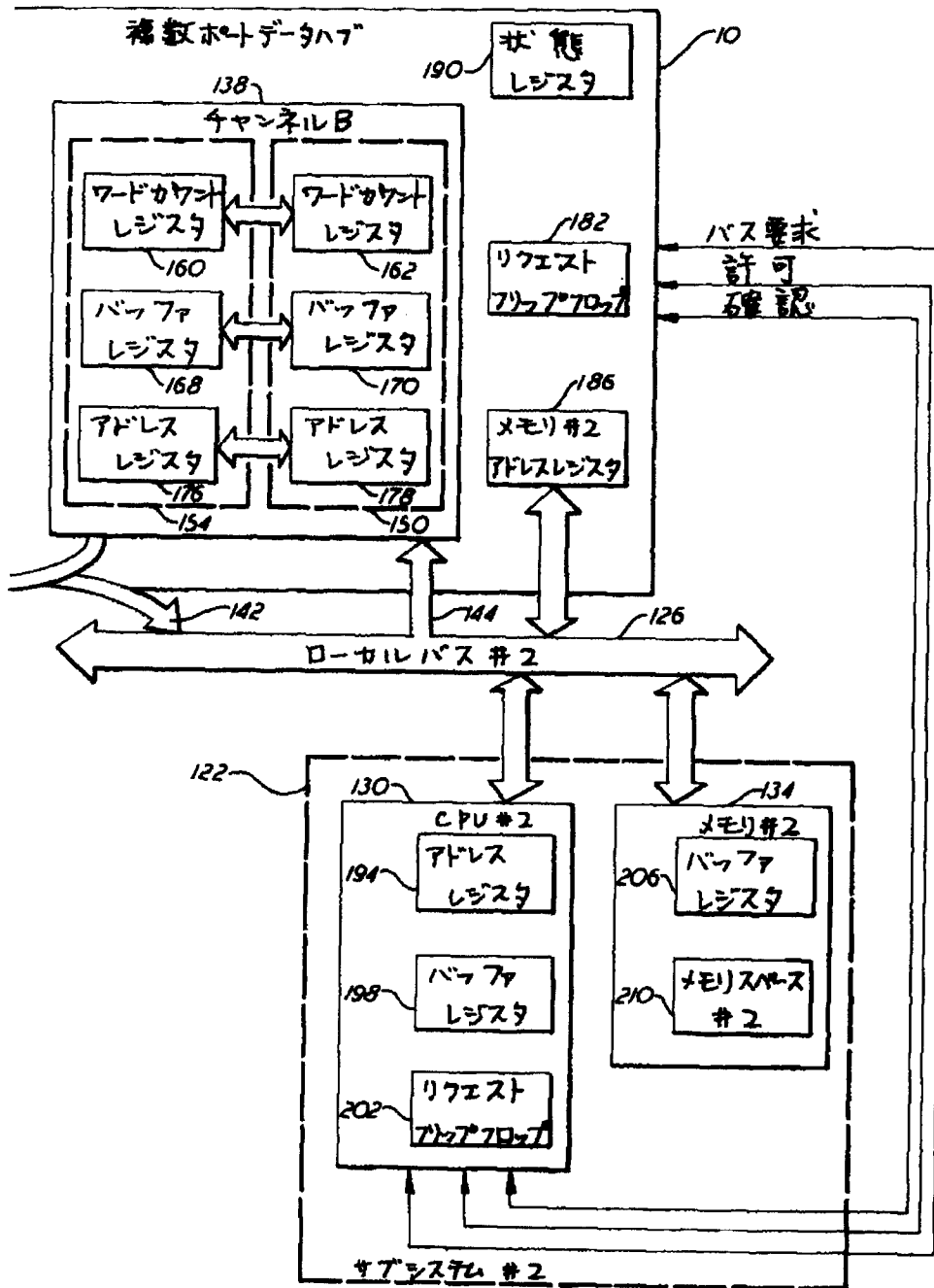
【第5図】



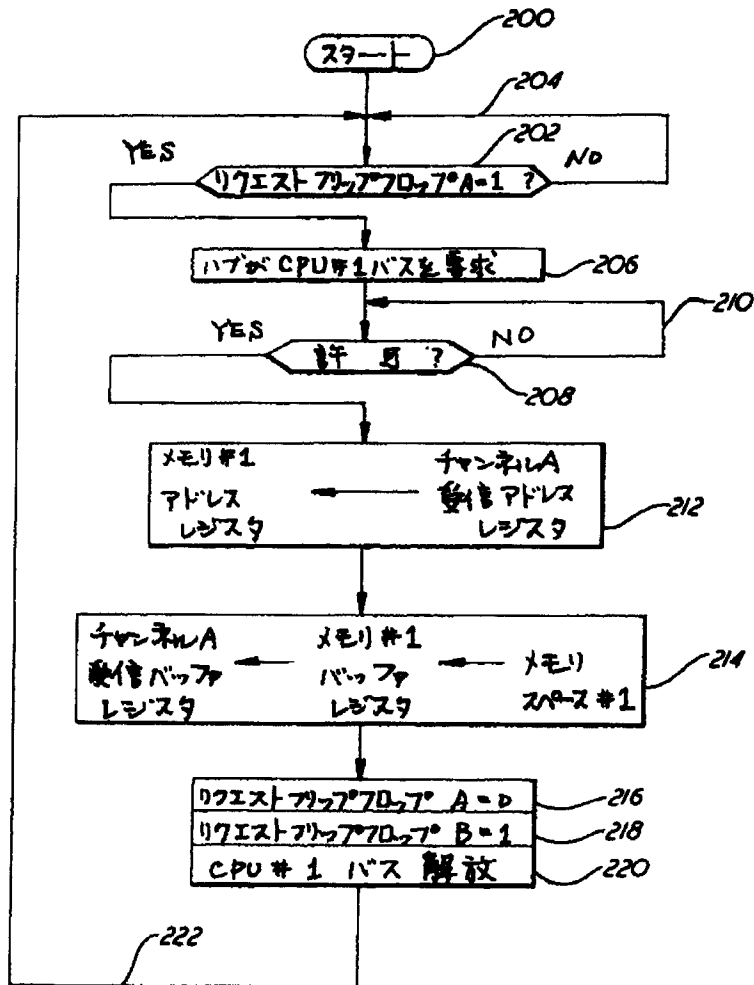
【第6A図】



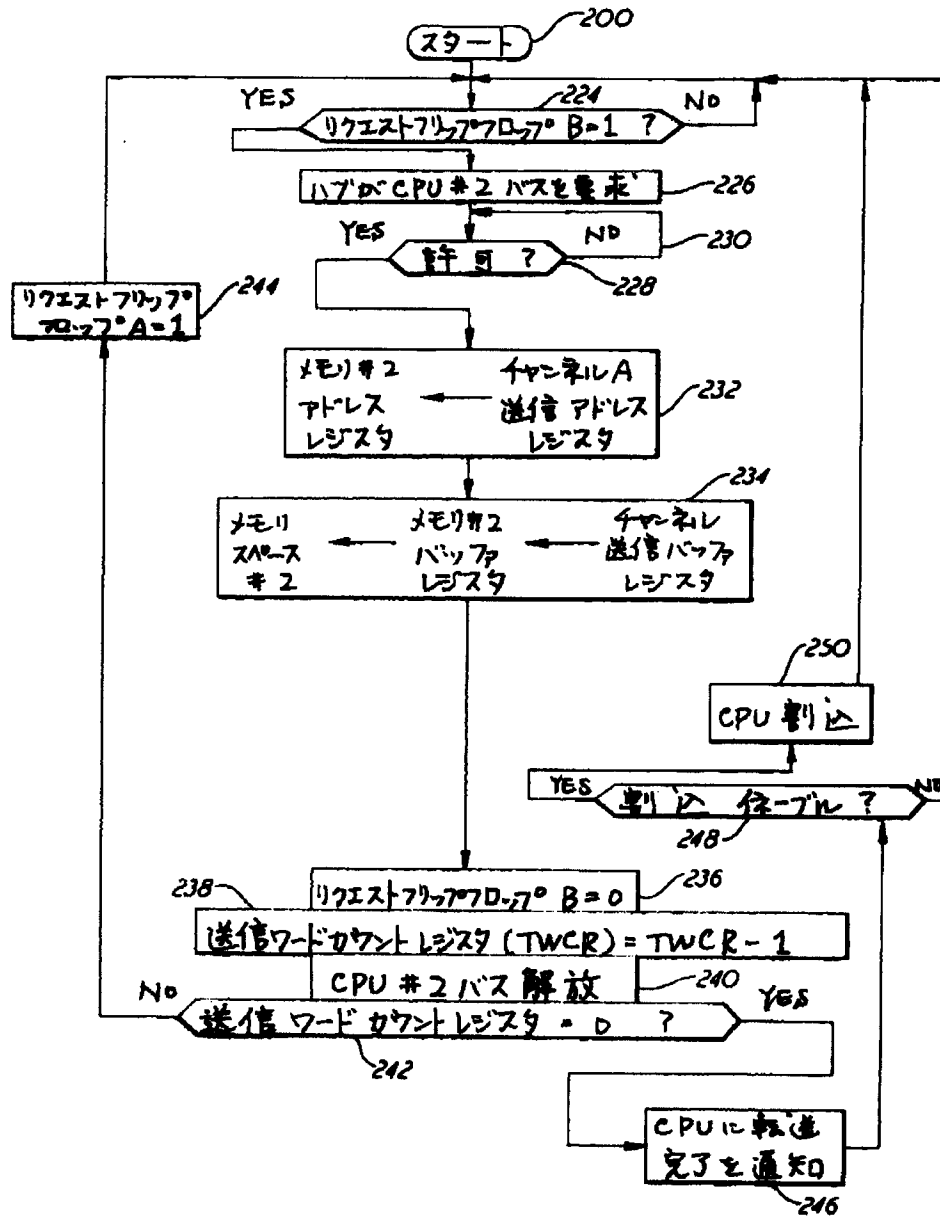
【第6B図】



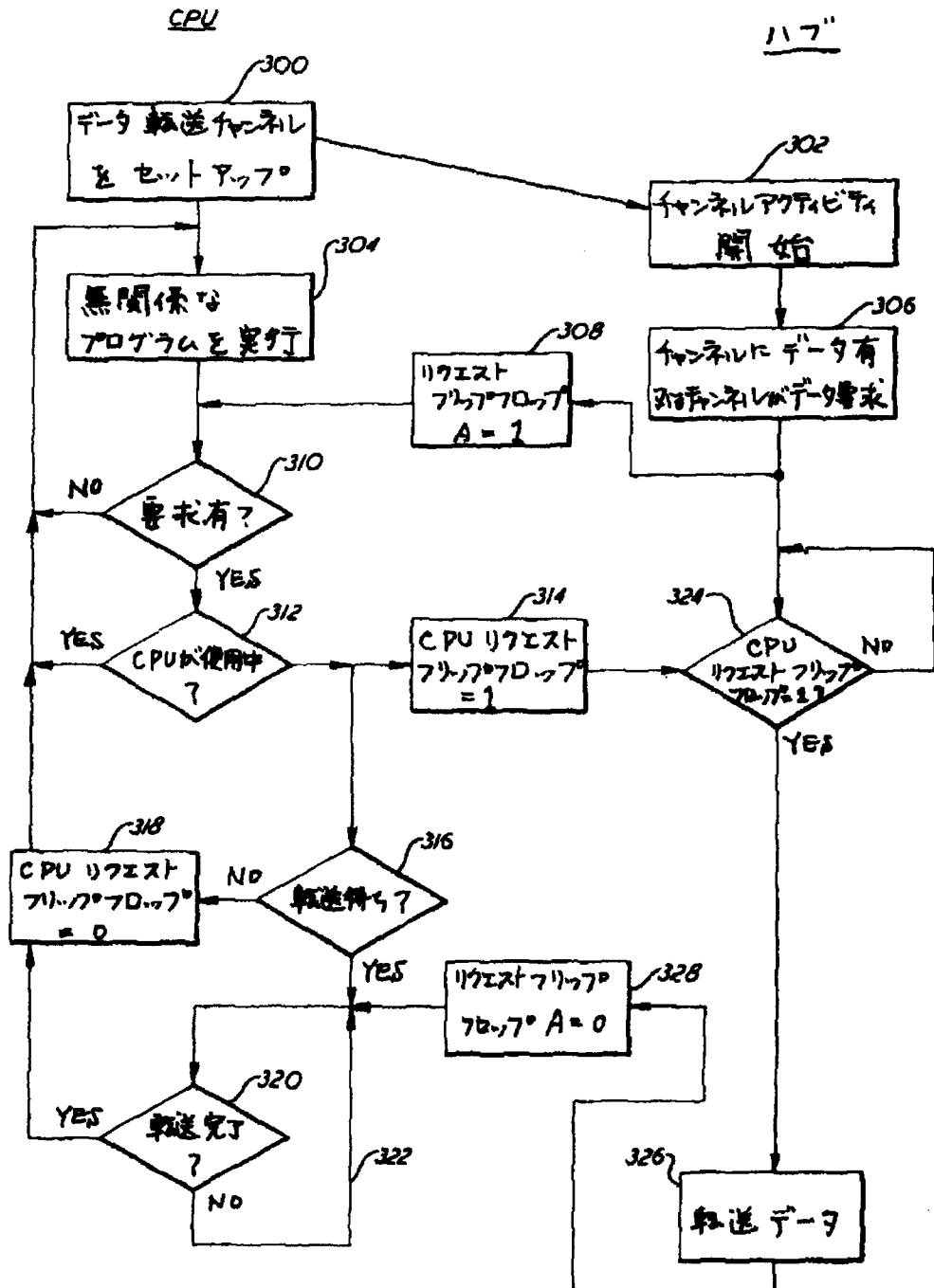
【第7A図】



【第7B図】

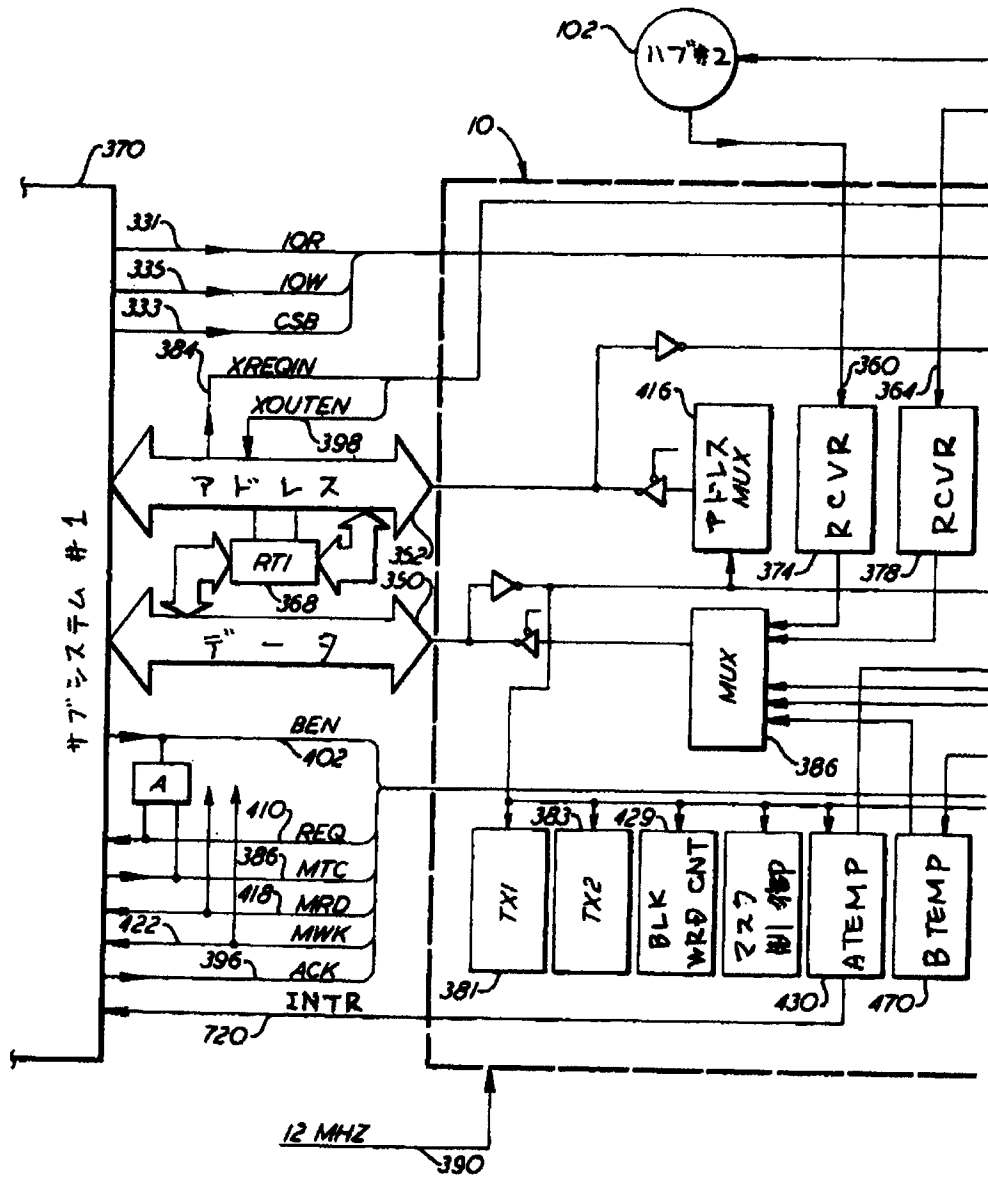


【第8図】

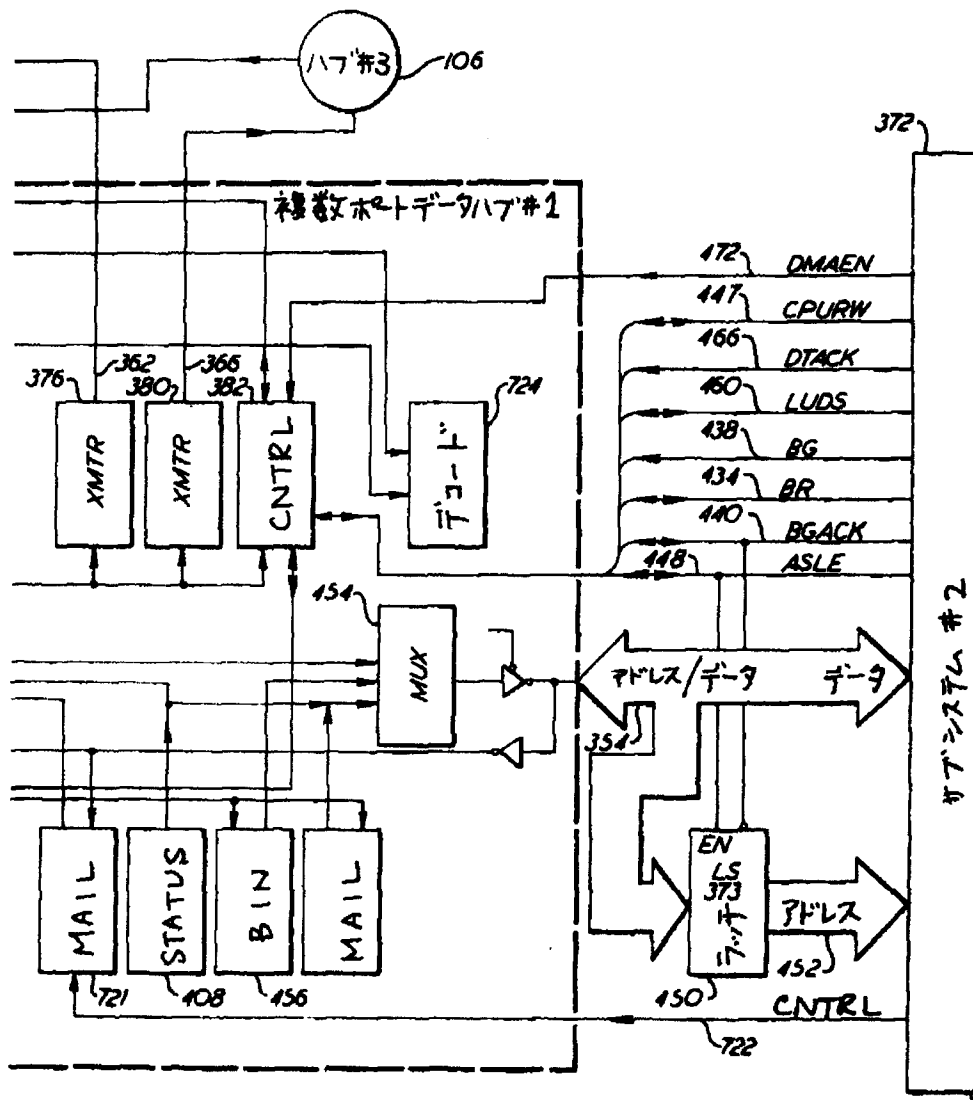




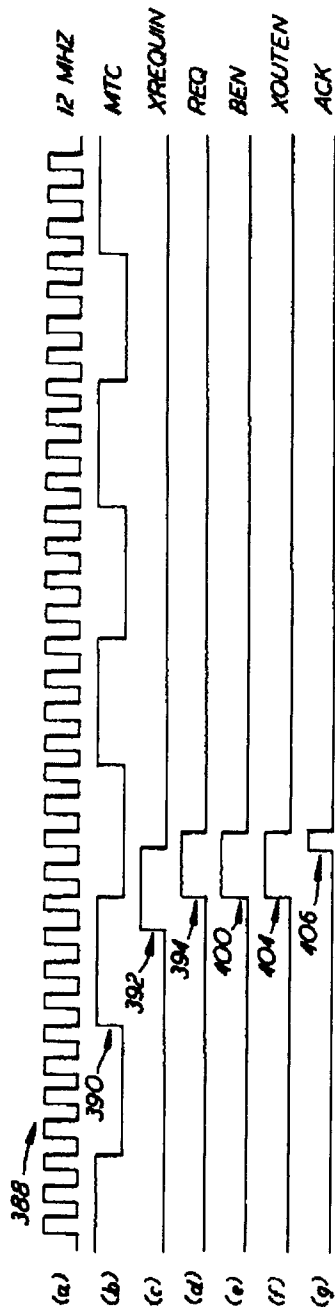
【第9A図】



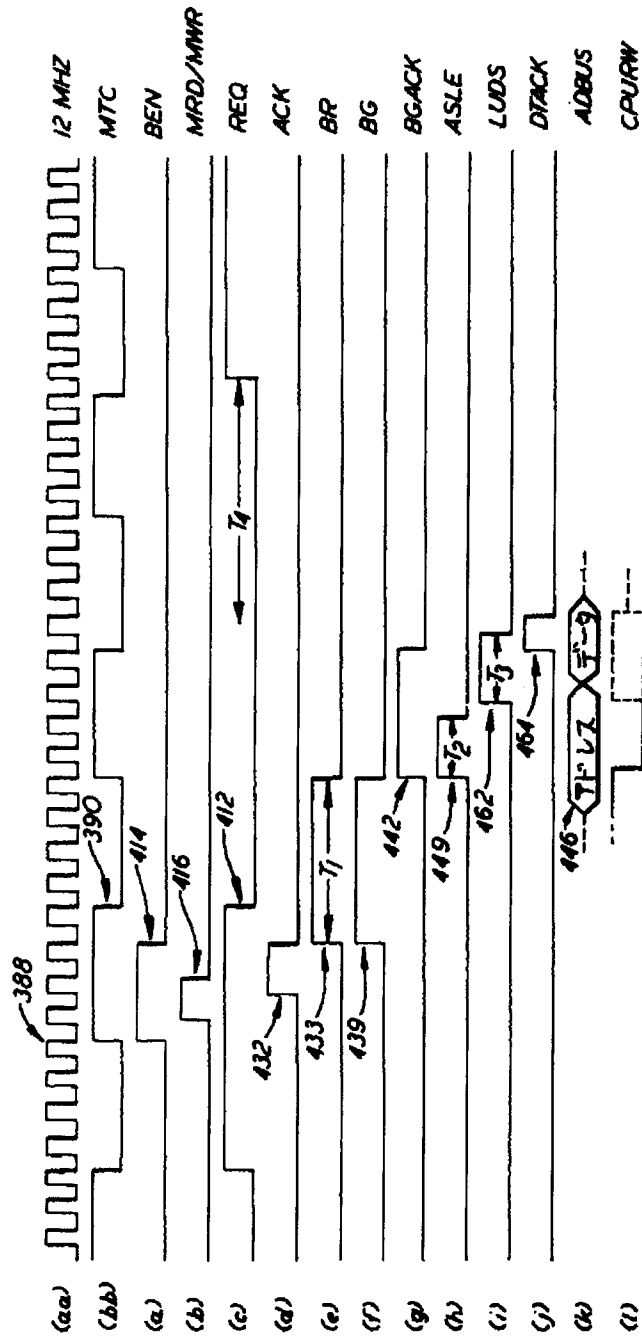
【第9B図】



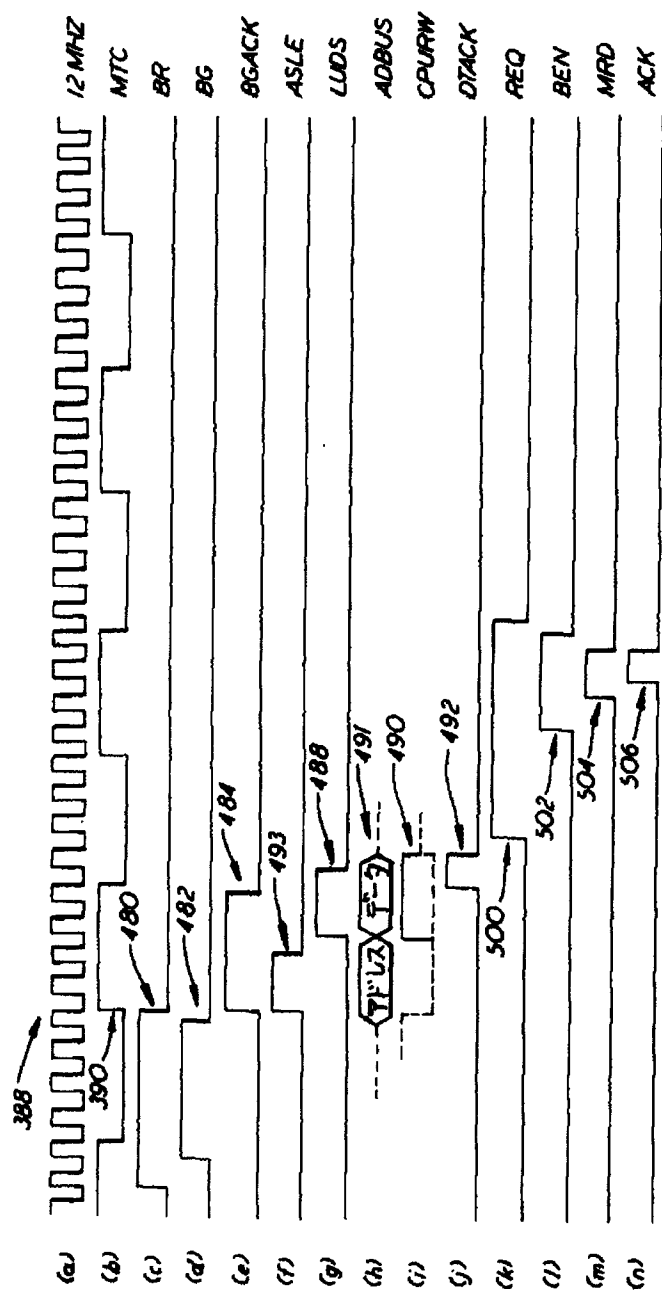
【第10図】



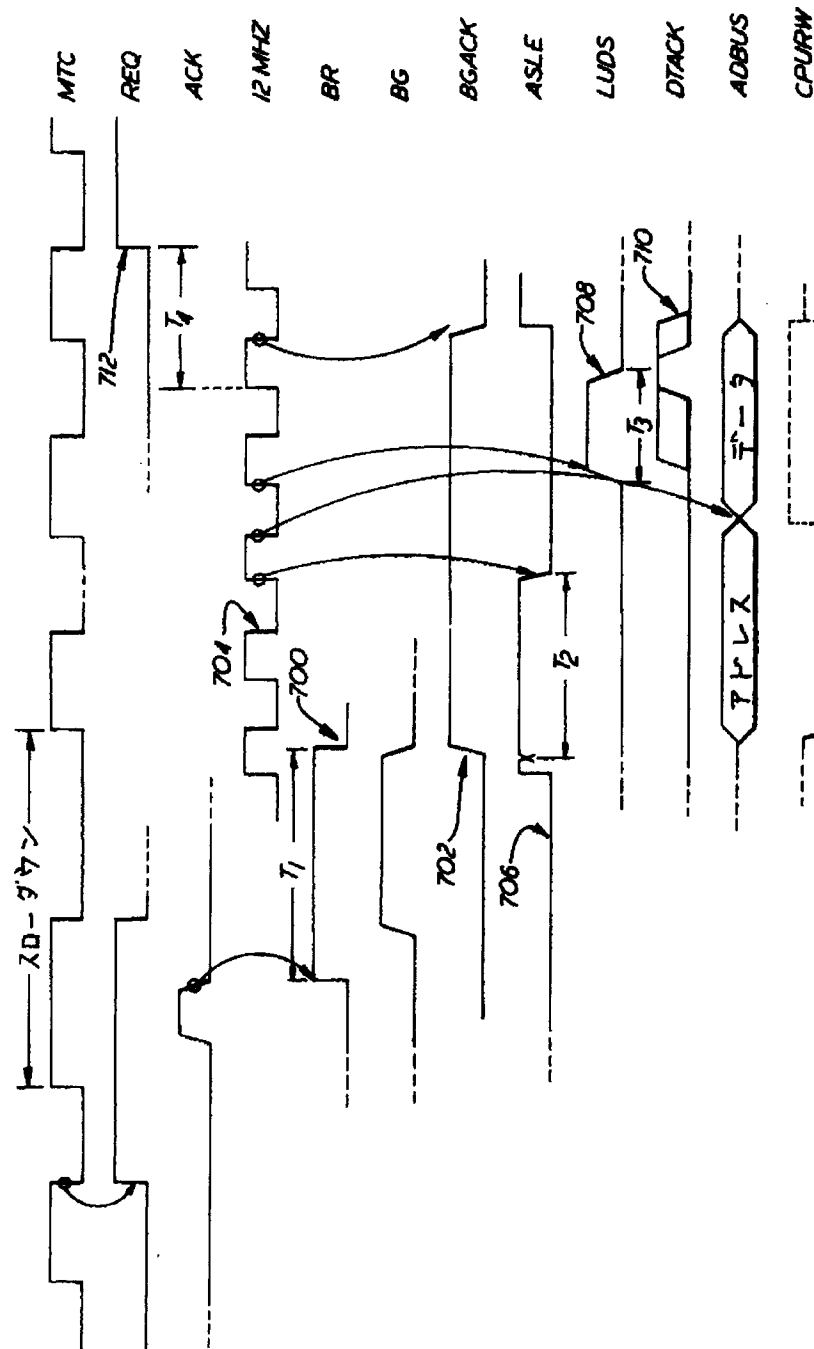
【第12図】



【第13図】



【第14図】



【第15A図】

下 位 バイト

読取書込  
デコード ↓

	08	07	06	05	04	03	02	01	
WR 0									Rx1 ADD PNTR
WR 1									Rx2 ADD PNTR
WR 2									Tx1 ADD PNTR
WR 3				カウント					Tx1 WRD CNT
WR 4									Tx2 ADD PNTR
WR 5				カウント					Tx2 WRD CNT
WR 6									SS1 BLK ADD PNTR
WR 7									SS2 BLK ADD PNTR
WR 8				カウント					BLK WRD CNT
WR 9	LS2 EN	LS1 EN	CH6 BXFR	CH5 TX2	CH4 TX1	CH3 RCV2	CH2 RCV1	CH1 XREQ	MASK & ENBL WORD
WR 10									LOAD Tx1 DIRCT
WR 11									LOAD Tx2 DIRCT
WR 12									XFR MM to SS2
WR 13									WRITE SS 1to2 MLBX
RD 0									READ Rx1
RD 1									READ Rx2
RD 2	MLBX DATA AVLB	DIR BIT	BWC >0	XMTR 2 WC>0	XMTR 1 WC>0	Rx2 BSY	Rx1 BSY	MLBX DATA AVLB	READ STATUS WORD
RD 3									READ MLBX 2to1
RD 4	LS2 EN	LS1 EN	CH6 BXFR	CH5 TX2	CH4 TX1	CH3 RCV2	CH2 RCV1	CH1 XREQ	MASK & ENBL WORD
RD 5									READ MLBX 1to2
WR 14									WRITE 2to1 MLBX

読取書込  
デコード

上位バイト

[illegible]

フロントページの続き

(72)発明者    グレゴリー   アラン   ポータノバ  
                 アメリカ合衆国   コネティカット  
                 06082   エンフィールド   アーバー   ロ  
                 ード   50番地

(56)参考文献    特開   昭53-33555 (J P, A)

                 特開   昭59-11053 (J P, A)